ADA034993

AFAL-TR-76-247

12

PROFGEN - A COMPUTER PROGRAM FOR GENERATING FLIGHT PROFILES

REFERENCE SYSTEMS BRANCH
RECONNAISSANCE AND WEAPON DELIVERY DIVISION

NOVEMBER 1976

TECHNICAL REPORT AFAL-TR-76-247
FINAL REPORT FOR PERIOD JUNE 1975 - FEBRUARY 1976

Approved for public release; distribution unlimited

AIR FORCE AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

STANTON H. MUSICK, Engineer

FOR THE COMMANDER

RONALD L. RINGO, Acting Chief
Reference Systems Branch
Reconnaissance & Weapon Delivery Division

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE - 22 DECEMBER 76 - 100

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

14

1. [redacted]

AFAL-TR-76-247

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

6

4. TITLE (and Subtitle)

PROFGEN - A Computer Program for Generating Flight Profiles.

9

5. ~~TYPE OF REPORT & PERIOD COVERED~~

Final Rept.

Jun [redacted]75 - Feb [redacted]76

6. ~~PERFORMING ORG. REPORT NUMBER~~

7. AUTHOR(s)

10

Stanton H. Musick

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Reference Systems Branch
Reconnaissance and Weapon Delivery Division
Air Force Avionics Laboratory, WPAFB, OH 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

6095 0501

11. CONTROLLING OFFICE NAME AND ADDRESS

Same as 9

11

12. ~~REPORT DATE~~

Nov[redacted]r [redacted]76

13. ~~NUMBER OF PAGES~~

200

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

16 6095

17 05

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

12 199p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Trajectory driver	Numerical integration
Profile generator	Navigation
Computer simulation	Flight path
Six degree-of-freedom	Wander azimuth

Written in Fortran

The PROGRAM

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes ~~a~~ computer program ~~that~~ calculates flight path data for an aircraft moving over the earth. The program is called PROFGEN, is written in FORTRAN, and is intended to support simulations that require a six degree-of-freedom trajectory driver.

(Cont'd on reverse)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

011670

(cont fr p i)

PROFGEN computes the position, velocity, acceleration, attitude and attitude rate of an aircraft flying over an ellipsoidal earth and responding to maneuver commands specified by the program user. Four types of maneuver commands are available: vertical turn, horizontal turn, sinusoidal heading change and straight flight. In addition, a speed change may be superimposed on any maneuver. Extended flight paths are created by stringing together a sequence of maneuvers.

PROFGEN uses a fifth-order numerical integrator to solve the kinematic equations of motion. This high-order integrator can operate in a self-analysis mode to produce a highly consistent set of values for position, velocity, acceleration, etc. In addition to using such an integrator, PROFGEN insures self-consistent and accurate results by (1) adjusting the step size to suit the problem's dynamics, (2) using the exact non-linear differential equations of motion, (3) avoiding integrations that span abrupt rate changes and (4) stopping the integration process to make output only when required by the user.

PROFGEN was developed on a CDC CYBER-74 computer where it compiles in about six seconds and uses less than 60,000 words of memory. The program includes a plotting capability that increases the memory requirement to 137,000 when installed.

FOREWORD

This technical report was prepared by Stanton H. Musick of the Reference Systems Branch, Reconnaissance and Weapon Delivery Division, Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio.

This work was initiated under Project Work Unit Number 60950501 and spanned the period from June 1975 through February 1976. The final manuscript was typed by Mrs. Shirley Suttman and was originally released in March 1976 as AFAL-TM-76-3. N.H.

Since the initial release in March 1976, one minor sign correction has been made in the PROFGEN program (see Subroutine GRAVITY in the listing) while numerous revisions have been made in this manuscript to correct mistakes and improve its readability.

ACKNOWLEDGEMENTS

The author would like to recognize two people for their substantial contributions to the development of PROFGEN: Jay Burns for developing and documenting the equations necessary to maintain flight in a great circle plane, and for doing the analysis that lead to a companion program named HEADING (see page 35); and Dave Kaiser for the writing, debugging and testing of HEADING and of all the code that produces plotted output in PROFGEN.

The author would also like to thank Shirley Suttman for her patience and skill in typing this report.

CONTENTS

CONTENTS (Continued)

ILLUSTRATIONS

TABLES

NOTATION

Subscripts, Superscripts, Prefixes

$\triangleq$	Equals by definition
$\approx$	Equals approximately
$(\underline{\ })$	Physical vector
$(\underline{\ })^j$	Math vector with components in j frame
$(\ )^T$	Matrix or vector transpose
$(\dot{\ })$	Time derivative
$\Delta(\ )$	The change over time of the variable ( )
$(\bar{\ })$	Average value
C_j^k	Transformation matrix, frame j to frame k

Coordinate Frames

Frame	Symbol	Components
Inertial	i	X_i, Y_i, Z_i
Earth	e	X_e, Y_e, Z_e
Navigation	n	x, y, z
Cardinal navigation	-	N, W, U
Path	p	x_p, y_p, z_p

I. INTRODUCTION

This report describes a computer program that calculates flight path data for an aircraft moving over the earth. The program is called PROFGEN and was written in FORTRAN. Its primary intended use is to support simulations that require a six degree-of-freedom trajectory driver.

This version of PROFGEN evolved from one written in 1973 that became obsolete because it lacked a wander-azimuth capability and employed an unrealistic roll control mechanization. These shortcomings are corrected in the revised version and several new features are added including output at user-determined times, the computation of attitude rates, an improved gravity model and the ability to turn through a precise angle without overshoot. In addition the revised version is coded in a modular fashion for ease of understanding and change.

This report will document PROFGEN in full. Section II is a general description of PROFGEN's capabilities and limitations that should allow the reader to determine the program's applicability to his problem. Section III is a user's guide that tells how to construct a flight profile with the available input parameters. Section IV develops the equations that PROFGEN solves. Section V describes the program itself. Appendix A presents an example problem and Appendix B gives a listing of the program source deck.

II. GENERAL CHARACTERIZATION

PROFGEN computes position, velocity, acceleration, attitude and attitude rate for an aircraft moving over the earth. Position is given as (geographic) latitude, longitude and altitude (see Figure 1). Velocity with respect to earth is componentized and presented in a local-vertical frame (x-y-z in Figure 1) that will be called the navigation frame. Acceleration consists of velocity rates-of-change summed with Coriolis effects and gravity. Attitude consists of roll, pitch and yaw, the Euler angles between the path frame and the navigation frame. These quantities will be defined precisely in Section IV.

Although the descriptions herein always refer to "aircraft" flight paths, PROFGEN has applicability to path generation for land and sea craft as well. In general PROFGEN is suited for simulation of any craft under continuous control. It is not well suited to describing bodies in free fall or earth orbit where mass attraction is the primary forcing function.

PROFGEN models a point mass responding to maneuver commands specified by the user. These maneuvers are available:

- vertical turns (pitch up or down)
- horizontal turns (yaw left or right)
- sinusoidal heading changes (oscillates left and right)
- straight flights (great circle or rhumb line path)

N-W-U	~	Geographic Coordinates
x-y-z	~	Navigation Coordinates
λ	~	Longitude
ϕ	~	Latitude (geographic)
h	~	Altitude

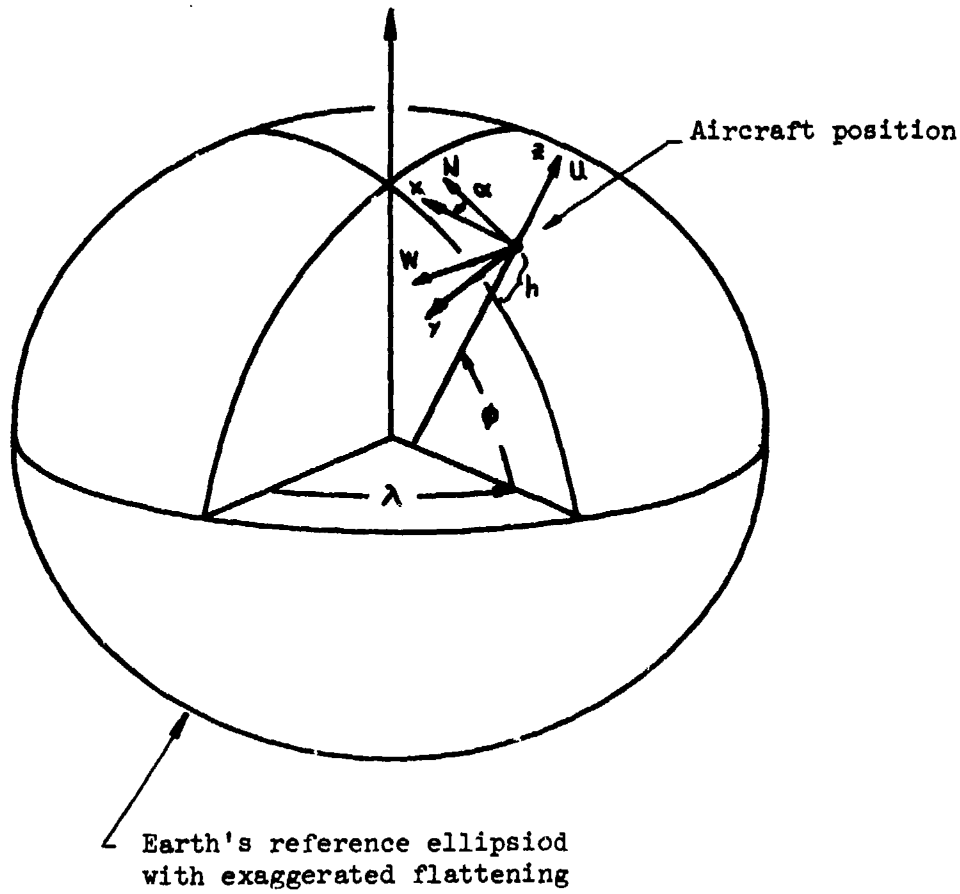


Figure 1 - Coordinate Frame Geometry

All horizontal-plane maneuvers are executed in a coordinated fashion. This simply means that the aircraft is rolled to an angle where the vector sum of the centrifugal turning force and the force of "gravity" (32.2 ft/sec^2) acts perpendicular to the wings. Only one type of maneuver may be executing at any given time but it can commence from any aircraft attitude. For example, the aircraft may go into a left turn while in a dive.

In addition to the four basic maneuvers, the user also has control of path acceleration by which the aircraft can be forced to change speeds. Path acceleration may be superimposed over any maneuver. This would allow, for example, an accelerated diving turn.

PROFGEN is used to create an extended flight profile by stringing together a sequence of maneuvers chosen from the basic four. The user specifies how long each maneuver shall last and thereby divides the profile into flight segments. Up to fifty flight segments may be strung together to produce a varied total profile. The final values of the variables in each segment are passed along as the initial values for the start of the next segment thereby creating uninterrupted time histories for all output variables.

The program allows step changes to occur in displacement acceleration and in rotational velocity. This produces continuous time histories for displacement velocity and rotational position (roll, pitch, yaw) but results in infinite jerk (rate-of-change of displacement acceleration) and infinite rotational acceleration.

Acceleration, velocity and position are related instantaneously by integration and differentiation to within the accuracy of the Kutta-Merson numerical integrator. Every effort has been made to configure this integrator to produce an accurate result so that the output variables form a self-consistent set. Thus the integrator is fifth order and can adjust its step size automatically to control the growth of errors. To illustrate, a great circle path from Dayton to Moscow accumulated less than 15 feet of error over its 5000 mile distance.

PROFGEN is limited in its capability to simulate intricate fighter maneuvers. This arises in part because PROFGEN forces the aircraft body and path frames to be coincident and thereby loses the ability to simulate slipping or crabbing motion. Thus, for example, one could not simulate a fighter aircraft doing a barrel roll or an Immelmann. On the other hand, one could simulate a complete loop of arbitrary radius since severity of maneuver is not restricted. In general, PROFGEN can simulate any maneuver possible with a bomber or cargo aircraft.

The earth is modeled as a perfect ellipsoid having values for eccentricity, semimajor axis length, spin velocity and gravitational constant equal to those of the DOD World Geodetic System 1972 (Ref. 1). Earth's gravity is modeled as a function of latitude and altitude, having both radial and level components. This model is not overly precise (probably no better than 25 micro gees) and may need revision for some applications.

PROFGEN compiles and executes in less than 60,000 words of CDC CYBER-74 memory. It uses only single precision variables and all source code is FORTRAN. The program takes six seconds of central processor time to compile. The ratio of simulated time to execution time improves as problem dynamics become less severe, reaching 20267 : 1 for straight flight segments but falling to 4 : 1 for a 10 gee horizontal turn.

III. USER'S GUIDE

This section defines the input data that the user supplies to run PROFGEN. The input data specifies

- initial conditions
- maneuver characteristics
- integrator control
- output control

All data is entered under a NAMELIST format that permits the entry of character strings. A character string is a parameter name followed by its values written in the user's choice of format specification. The use of NAMELIST on the CDC CYBER-74 will be illustrated in Figures 2 and 3.

Two NAMELIST input data lists are used, PRDATA and PASDATA. The PRDATA (Problem Data) list contains 15 parameters that remain fixed for the entire run. These parameters specify all initial conditions and control output.

The PASDATA (Path Segment Data) list contains 13 parameters that remain fixed only for the length of a segment. These parameters specify and describe each maneuver, control the numerical integrator, and control the output frequency.

3.1 PRDATA Input

Fifteen parameters are entered through the PRDATA list. Failure to specify any one of these results in program termination. All

parameters are single precision and all must be entered in units of feet, seconds and/or degrees. The following format will be used to describe input parameters throughout this section and the next.

Parameter	(Type)	Units (If Any)
IPROB	(Integer)	

The problem identification number. It is set by the user for identification purposes only.

NSEGT (Integer)

The total number of path segments required to complete the entire problem. This number may not exceed 50 as the program is now configured.

LLMECH (Integer)

The local-level azimuth angle mechanization index. See Section 4 and Table 2.

LLMECH	Azimuth Mechanization
1	Alpha Wander
2	Constant Alpha
3	Unipolar
4	Free Azimuth

TSTART (Real) seconds

The initial time. It is used to begin the problem at any desired point. It may be negative.

VTO	(Real)	feet per second

The initial magnitude of total velocity with-respect-to the earth. VTO must be non-negative.

PHEADO	(Real)	degrees

The initial heading angle of the path coordinate frame. It is specified as positive clockwise from North. Its range is the closed interval [-180., +180.].

PPITCHO	(Real)	degrees

The initial pitch angle of the path coordinate frame. It is specified as positive in the upward direction. The path frame is level when the pitch angle is zero. Its range is [-90., +90.].

ALFAO	(Real)	degrees

The initial alpha angle. Alpha is the navigation frame heading angle and is specified positive counterclockwise from North. Its range is [-180., +180.].

LATO	(Real)	degrees

The initial geographic latitude. Its range is the open interval (-90., +90.). Since the program falters when trying to compute at exactly 90 degrees, these two extreme points must be avoided.

LONO | (Real) | degrees

The initial longitude. It has no effect on the problem's dynamics but is necessary to establish a reference point for the calculation of current position. Its range is [-180., +180.].

ALTO | (Real) | feet

The initial altitude above the reference ellipsoid. ALTO may be negative.

IPRNT | (Integer)

Print control index having control, in part, over what is written on TAPE6. This tape is considered to be printed output. All TAPE6 output is formatted.

IPRNT	Action
1	Output on TAPE6 at time-intervals specified by DTO (a PASDATA parameter)
≠1	Output at DTO intervals is turned off.

Regardless of the state of IPRNT, the following output also appears on TAPE6:

- date and time
- input data from PRDATA and PASDATA lists
- variable values at start of each segment and at t-final
- error messages
- post-run assessment of numerical integrator performance

IRITE (Integer)

Write control index. This output is written on TAPE3 and is designed for compact storage of data for subsequent use by another program. All TAPE3 output is unformatted.

IRITE	Action
1	Output on TAPE3 consisting of date, time, input data and variable values beginning at TSTART and continuing at DTO intervals.
≠1	No output on TAPE3.

IPLOT (Integer)

Plot control index. This output is on PLFILE for post-run graphing using DISSPLA, a CALCOMP plot library.

IPLOT	Action
1	Program plots five graphs, latitude vs. longitude and time histories of altitude, roll, pitch and yaw. Up to 501 points are plotted in each graph, the first being at TSTART and all thereafter at DTO intervals.
≠1	No plotted output.

ROLRATE (Real) degrees per second

Nominal aircraft roll rate. When the aircraft must bank to execute a coordinated horizontal turn, it rolls to the proper bank angle at a rate of ROLRATE. In sine-heading-change maneuvers, ROLRATE serves as the limiting value for the derivative of roll. ROLRATE must be positive.

Figure 2 is a sample of a PRDATA card input set. Note that the data items may be listed in any order so long as they all appear between the beginning identifier, $ PRDATA, and the ending identifier, $.

3.2 PASDATA Input

Thirteen parameters having up to 50 values each are entered through the PASDATA list. Each parameter is dimensioned in the program as a 50 element array, the number 50 corresponding to the maximum number of segments allowed. Each parameter value must be assigned to the array element corresponding to its segment number; for example, if the output spacing in the sixth segment is to be 25 seconds, one would input DTO(6) = 25. Each parameter name in the list that follows has the argument i appended to it to indicate its dependence on segment i, $1 \leq i \leq 50$.

Six of the PASDATA parameters (TURN, NPATH, PACC, TACC, HEAD, PITCH) describe the maneuver and four (MODE, ERROR, HMAX, HMIN) are associated with numerical integration. The other three control output frequency (DTO), set segment length (SEGLNT), and control initial conditions (RESTART). Each parameter has a default option that is invoked in lieu of input data. The default saves the user the trouble of specifying values that often recur. All parameters are single precision and all must be entered in units of feet, seconds, gees (1 gee $\triangleq$ 32.2 ft/sec.^{2}) and/or degrees.

```
$PRDATA IPROB=650,
NSEGT=17,
LLMECH=2,
TSTART=0.,
VTO=1000.,
PHEADO=180.,
PPITCHO=0.,
ALFAO=45.,
ALTO=30000.,
LATO=39.,
LONO=-84.,
ROLRATE=250.,
IPRNT=1,
IRITE=0,
IPLOT=1$
```

Figure 2 - Sample of PRDATA Input

Parameter	(Type)	Units (If Any)
SEGLNT(i)	(Real)	seconds

The time interval of the i^{th} segment. SEGLNT(i) can be any non-negative number, including zero. The program remains in segment i until exactly SEGLNT(i) seconds have been simulated. The default value is zero seconds.

RESTART(i) (Integer)

The index number for control of the initial conditions at the beginning of each segment.

RESTRART(i)	Action
1	All variables in the state vector are reset to the conditions that existed at TSTART, namely those in PRDATA. RESTART = 1 is useful when one wishes to produce a reference flight, and a variation of that flight, all in one run.
≠1	The variable values at the beginning of segment i equal those at the end of segment i-1.

The default value is zero, no reset performed.

TURN(i)

The index number for the type of maneuver to be used.

TURN(i)	Action
1	vertical turn
2	horizontal turn
3	sinusoidal heading change
4	straight flight

All maneuvers begin at the start of a segment. Vertical and horizontal turns are complete when a specified turn angle is reached. If specified angle is reached and time remains in the segment, PROFGEN reverts to a straight flight mode (TURN = 4) for the remaining seconds of the segment. If TURN(i) is 3, a "sinusoidal" path (oscillatory yawing motion in the horizontal plane) is flown for SEGLNT(i) seconds. For sine maneuvers, the user must select a segment length that is a multiple of Tp/4 where Tp is the period of the sinusoid. If TURN(i) is 4, a straight-flight segment will be flown over a nominal path determined by the value of NPATH(i) for SEGLNT(i) seconds. Section 3.4 discusses these maneuver characteristics more fully. The default value is 4, straight flight.

NPATH(i) (Integer)

The index number for the nominal path.

NPATH(i)	Action
1	Great cricle path
2	Rhumb line path

When a rhumb line path is chosen, the aircraft maintains a constant heading angle during straight flight periods. When a great circle path is chosen, the aircraft flies in a fixed plane during straight flight periods. The aircraft maintains this fixed-plane flight over the ellipsoidal earth, even when altitude changes, by correcting heading continuously. When not in straight flight (i.e. TURN = 1, 2 or 3), the rhumb line or great circle actions are superimposed on the chosen maneuver. The default value is 2, rhumb line path.

PACC(i) (Real) gees

The signed value of the constant acceleration along the velocity vector, i.e. along the path x-axis. The program converts PACC(i) in gees to path acceleration in feet/second^2 by multiplying by 32.2. PACC(i) may be assigned any real

value; it remains that value for the entire segment regardless of maneuver specification. Positive (negative) values cause the aircraft to gain (lose) total speed. Since all active maneuvers (TURN = 1, 2 or 3) require a division by total speed (VT) to compute acceleration, the user must assign PACC(i) so VT is never zero during the actual turning portion of such maneuvers. PACC(i) may force VT to zero anytime during a straight flight segment. The default is zero gees.

TACC(i) (Real) gees

The magnitude of the maximum centrifugal acceleration during either a vertical or horizontal turn. The program converts TACC(i) in gees to acceleration in feet/second^2 by multiplying by 32.2. TACC(i) must be positive for vertical and horizontal turns. The default value is zero gees.

HEAD(i) (Real) degrees

HEAD(i) has two uses.

For horizontal turns, HEAD(i) is the desired change in heading angle. Other factors permitting (SEGLNT, TACC, ROLRATE, PACC, VT) this turn angle will be executed accurately. The magnitude of HEAD(i) may be greater than 360 degrees. A positive (negative) HEAD(i) forces a right (left) turn.

For sine maneuvers HEAD(i) is the maximum variation of the heading angle and its absolute value must be less than 90 degrees. A positive (negative) HEAD(i) forces the sine maneuver's ground track to lie right (left) of the initial ground track. The default value is zero degrees.

PITCH(i) (Real) degrees or deg/sec

PITCH(i) has two uses.

For vertical turns PITCH(i) is the desired change in pitch angle in degrees. Other factors permitting (SEGLNT, TACC, PACC, V_T), this value will be achieved precisely. PITCH(i) may exceed 90 degrees. A positive (negative) PITCH(i) forces the pitch angle to increase (decrease).

For sine maneuvers, PITCH(i) is the frequency of the sinusoidal rate of change of heading in degrees per second. It must be non-zero. The sign of PITCH(i) has no effect on the sine maneuver. The default value is zero in degrees or degrees per second, as the case may be.

DTO(i) (Real) seconds

The time interval between required output times. DTO(i) is referenced to zero seconds; e.g., if DTO(i) = 6, output would be available at T = (···, -12, -6, 0, 6, 12,···). DTO(i) must be positive. DTO(i) controls output frequency for printing, writing and plotting (see IPRNT, IRITE, IPLOT). Careful sizing of DTO(i) is a necessity, especially when two or three output modes are used simultaneously. The default value is 100 million seconds corresponding to no output at all.

MODE(i) (Integer)

The index for control of step size in the numerical integration routine.

MODE(i)	Action
0	Fixed step-size integration.
1	Variable step-size integration.

The step size is HMIN(i) when fixed step-size integration is used. A fifth order numerical integration is performed.

With the variable step-size mode, the program begins the integration with a step size of HMIN(i). The numerical integrator adjusts the step size upwards from there while keeping the within-step error below the value specified in ERROR(i). If problem dynamics are mild, the step size can grow very large, limited finally by HMAX(i). If problem dynamics are severe, the minimum step size may not be adequately small to satisfy the error criterion in which case an error message is printed.

In summary both integration modes perform fifth order numerical integrations but MODE = 1 adjusts step size automatically to conform to an error criterion. The default value is variable step-size integration.

ERROR(i) (Real)

The allowable within-step integration error. It must be positive. The default value is 10^{-6}, a value that has proven satisfactory during program development.

HMAX(i) (Real) seconds

The maximum step size when variable step-size integration is used. It must be positive. The default value is 10,000 seconds.

HMIN(i) (Real) seconds

The minimum step size when variable step-size integration is used. With fixed step-size integration, HMIN(i) is the size of each step. It must be positive. The default value is one second.

Table 1 shows the relationship of TURN, TACC, HEAD and PITCH. Figure 3 is a sample of a PASDATA card input set. Note that some parameters are not specified because the desired values agreed with the default option. Also note the capability to specify repeated values using a repetition factor.

```
$PASDATA
SEGLNT(1)=20.,30.,10.,30.,30.,40.,10.,10.,50.,10.,10.,10.,50.,10.,40.5,
50.,40.,
TURN(1)=4,3,4,3,2,2,1,2,4,2,1,4,2,4,2,4,2,
NPATH(1)=17*1,
TACC(5)=1.,1.,0.5,5.,0.,5.,0.5,0.,4.,0.,2.,0.,2.,
PACC(7)=-.1,
PACC(11)=.1,
P  C(17)=1.,
HEAD(1)=0.,20.,0.,-20.,-30.,30.,0.,-90.,0.,-90.,0.,0.,365.,0.,-135.,0.,
135.,
PITCH(1)=0.,36.,0.,36.,0.,0.,5.,3*0.,-5.,
MODE(1)=17*1,
HMIN(1)=17*.0001,
DTO(1)=17*1.$
```

Figure 3 - Sample of PASDATA Input

TABLE 1 DEFINITION OF TURN PARAMETERS+

	Vertical Turn	Horizontal Turn	Sinusoidal Heading Change	Straight Flight
TURN(i)	1	2	3	4
TACC(i)	Magnitude of vertical turn centrifugal acceleration.	Magnitude of horizontal turn centrifugal acceleration.	Not used.	Not used.
HEAD(i)	Not used. Heading will change slowly if great circle path selected.	Change in heading angle (+~CW).	Amplitude off nominal of sinusoidal flight path.	Not used. Heading will change slowly if great circle path selected.
PITCH(i)	Change in pitch angle (+~up).	Not used. Pitch remains unchanged.	Frequency of heading rate of change.	Not used. Pitch remain unchanged.

+See text for units

3.3 Program Limitations (What Happens If ...)

PROFGEN will not begin profile generation until each parameter lies within its permitted range as specified in 3.1 and 3.2. Subroutine VALDATA range-checks NSEGT, LLMECH, VTO, PHEADO, PPITCHO, ALFAO, LATO, LONO, ROLRATE, SEGLNT, TURN, NPATH, TACC, HEAD, PITCH, DTO, MODE, ERROR, HMAX and HMIN. A message is printed for each range-check that fails and the program is terminated.

Error messages can also occur during profile generation (i.e. after TSTART). One such mid-run message occurs if and when the integrator reduces step size to HMIN and is still not able to satisfy the error criterion (ERROR). In such cases this message is printed:

THE INTEGRATION ERROR EXCEEDS ITS ALLOWED VALUE

When this occurs PROFGEN is designed to continue to run, doing the best it can with HMIN. The value of the result is questionable, however, and the best advice is to scrap the output, reduce HMIN by at least a factor of ten, and rerun the program.

Another mid-run error message occurs if and when the product of computed roll rate and minimum step size would produce a roll bank angle in excess of 90 degrees. Since the aircraft must bank to execute either a horizontal turn or a sine maneuver, excessive roll angles could occur in either type of maneuver. PROFGEN avoids this problem in a horizontal turn but succumbs to it in a sine maneuver; prior to each sine maneuver the program checks for the problem and, if it exists, prints the following warning message and then terminates execution.

CHKSHC MESSAGE - THE PRODUCT OF COMPUTED
ROLL RATE AND MINIMUM STEP SIZE EXCEEDS
90 DEGREES. BANK ANGLES IN EXCESS OF
90 DEGREES ARE NOT ALLOWED. PROGRAM
TERMINATED.

Again the solution is to reduce HMIN for that segment.

Another mid-run message occurs if and when the cosine of pitch is exactly zero. This would happen, of course, if pitch magnitude were exactly $\pi/2$ radians (90 degrees). At 90 degrees, the algorithm for computing yaw rate and roll rate would make both of these quantities infinite. PROFGEN recognizes the situation and prints the following warning message from subroutine ETADOT.

ROLL AND YAW RATES ARE UNDEFINED
WHEN PITCH IS 90 DEGREES. THUS
ALL RATES HAVE BEEN TEMPORARILY
ZEROED.

No divisions by zero are attempted so the program continues to execute. In short PROFGEN handles a pitch angle of $\pm$ 90 degrees by avoiding the fatal rate computations.

If latitude becomes $\pm$ 90 degrees, PROFGEN attempts a division by zero in LAMDOT and suffers a fatal error in which the CDC operating system kicks the program off the machine. Similar zero-division failures occur when one attempts a horizontal plane maneuver (horizontal turn or sine maneuver) with horizontal velocity equal

zero, or when a vertical turn is attempted with total velocity equal zero, or when the aircraft is flown into the earth's center. Other zero-division situations would be even rarer than these and are not worth mentioning.

3.4 What to Expect from Each Maneuver

This section describes each maneuver in depth to see what it does and how it does it. These descriptions form the basis for the development of the control equations in Section 4.3.

3.4.1 Vertical Turn

A vertical turn is a pitch-up or pitch-down maneuver that takes place in a vertical plane. As with all maneuvers, vertical turns begin executing at the start of a segment (TI). Pitch angle advances, at a rate controlled by TACC and aircraft speed, until the time in the segment runs out at TF or until the change-in-pitch reaches PITCH degrees at TDONE, whichever time comes first. Altitude, pitch and acceleration curves for two vertical turns are shown in Figure 4.

Let a_n represent turn acceleration normal to the flight path. PROFGEN holds a_n (=TACC fps^2) constant while pitch advances. Since

$$a_n = \frac{V^2}{r} = V\dot{\theta} \qquad (1)$$

the turn's radius of curvature, r, and its advancement rate, $\dot{\theta}$, are also constant as long as total speed, V, remains fixed.

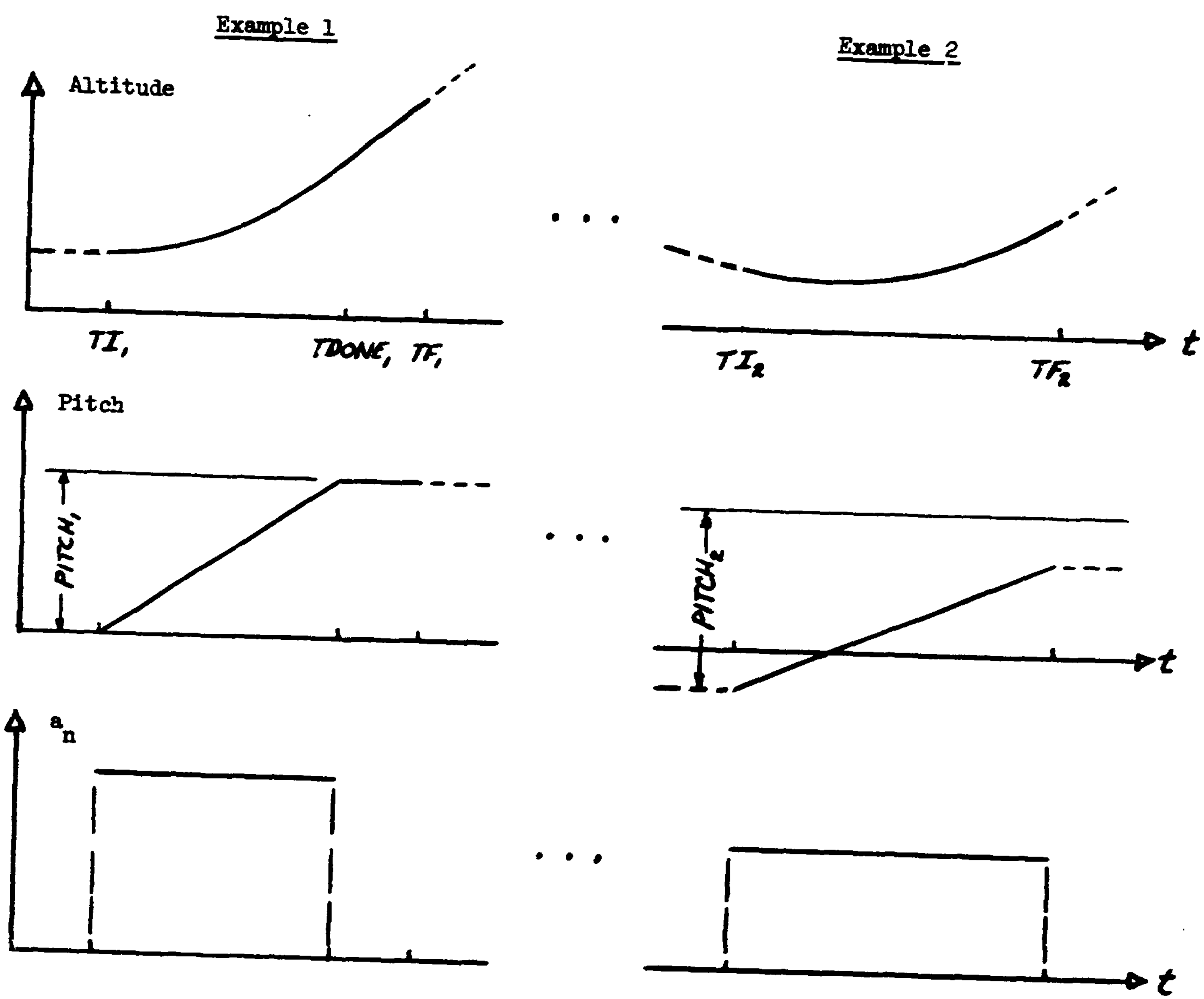


Figure 4 - Two Examples of Constant-Speed Vertical Turns

Turning action is enabled by switching $\dot{\theta}$ on at TI and then off at min (TF, TDONE). This produces a pitch-rate discontinuity at min (TF, TDONE) that the numerical integrator, KUTMER, cannot handle. PROFGEN solves the problem by splitting the segment into two pieces, one from TI to TDONE and the other from TDONE to TF. (If TDONE > TF, only one piece is necessary, viz. TI to TF.) KUTMER integrates the two disjoint pieces separately and thereby avoids a time step that would span the pitch-rate discontinuity.

The switching action on $\dot{\theta}$ may be observed in the program's pitch-rate output which is a non-zero constant while pitch is advancing and zero thereafter. Vertical plane maneuvers induce no rolling or yawing motion.

TDONE is computed in subroutine TSETUP1 before segment integration begins. The computation for TDONE assumes two things:

- turn acceleration is constant
- total speed does not drop to zero

The first assumption is guaranteed by the program's construction. The user must guarantee the second assumption by choosing PACC so total speed will remain positive. When these assumptions hold the aircraft's PITCH angle will advance exactly PITCH degrees in the interval TI to TDONE as illustrated in Example 1 of Figure 4. If TDONE exceeds TI, the change-in-pitch will fall short of PITCH as illustrated in Example 2 of Figure 4.

The minimum time required to complete a vertical turn through an arbitrary pitch angle $\Delta\theta$ is as follows:

$$\Delta t = \begin{cases} \dfrac{V_o \, \Delta\theta}{a_n} , & \dot{V}_o = 0 \\ \dfrac{V_o}{\dot{V}_o}\left(\exp\left(\dfrac{\dot{V}_o}{a_n}\Delta\theta\right) - 1\right) , & \dot{V}_o \neq 0 \end{cases} \tag{2}$$

where Δt = time required to pitch through $\Delta\theta$ radians (>0)

V_o = total speed at TI (>o)

$\Delta\theta$ = turn angle = |PITCH| (>o)

a_n = normal turning acceleration = TACC (>o)

$\dot{V}_o$ = tangential acceleration = PACC

A derivation of this result is given in Section 4.3.2. Equation (2) is useful for computing flight time in a pitch maneuver.

3.4.2 Horizontal Turn

In a horizontal turn the aircraft heading swings left or right to force the aircraft to follow a pseudo-circular path over the ground. Such a turn can be performed in any pitch attitude except ± 90 degrees. Horizontal turns are always performed in coordinated fashion. (Coordinated turns are also termed symmetric.) A coordinated turn is one in which the aircraft roll (bank) angle is controlled so that the vector sum of the horizontal turning force and the vertical force of "gravity" (defined for this purpose as 32.2 ft/sec^2) acts perpendicular to the wings. For example, in a level one-gee turn to the pilot's right, the aircraft rolls about its long axis to a bank angle of 45 degrees, right wing down. Because heading and roll must both be controlled, the software implementation for the horizontal turn is more complex than that for the vertical turn.

As was true with pitch in the vertical turn, heading advances in the horizontal turn until the time in the segment runs out at TF or until the change-in-heading reaches HEAD degrees at TDONE, whichever time comes first. Another way to say this is that the aircraft turns in the time interval between TI and min (TF, TDONE). During this turning interval, while heading advances continuously, roll also goes through its own set of gyrations in order to implement a coordinated turn. Representative roll curves are shown in Figure 5.

Note that roll always begins and ends at zero and remains in the interval $(-90^\circ, +90^\circ)$. Also note that when roll changes, it does so at the constant rate, ROLRATE.

In contrast to the vertical turn where a_n was constant, a_n for the horizontal turn follows a curve similar in shape to the roll curves from Figure 5. a_n is given by

$$a_n(t) = 32.2 \cos(\eta_y) \tan(\eta_x(t)) \qquad (3)$$

where η_y is (constant) pitch and η_x is roll. Note that, since η_x varies with time, a_n does also thereby producing a path with a variable radius of curvature. (The radius of curvature is infinite at the two ends of the turn and reaches a minimum when bank angle peaks.) Lat-long, yaw, roll and acceleration curves for two horizontal turns are shown in Figure 6.

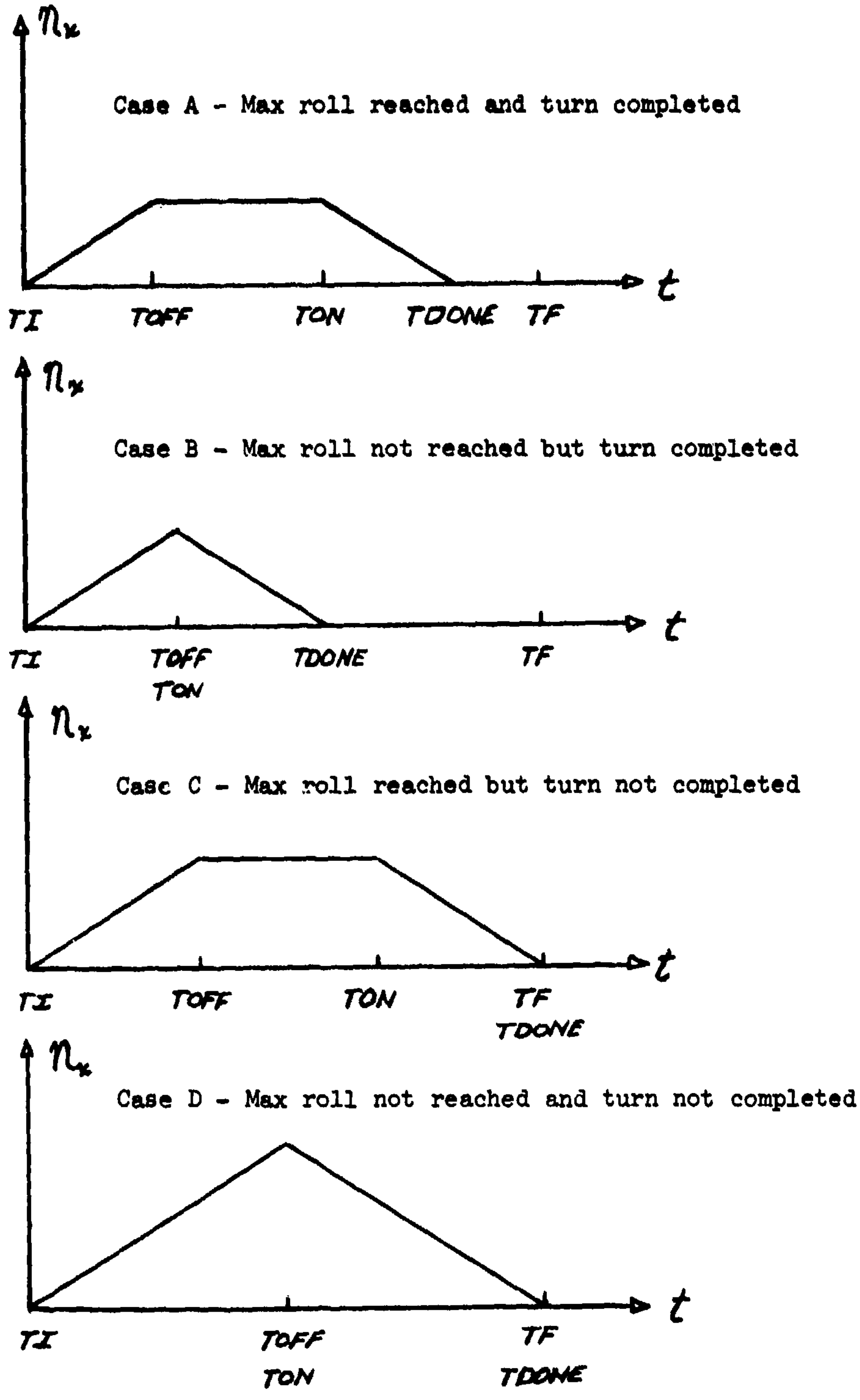


Figure 5 - Roll Angle Behavior in a Horizontal Turn

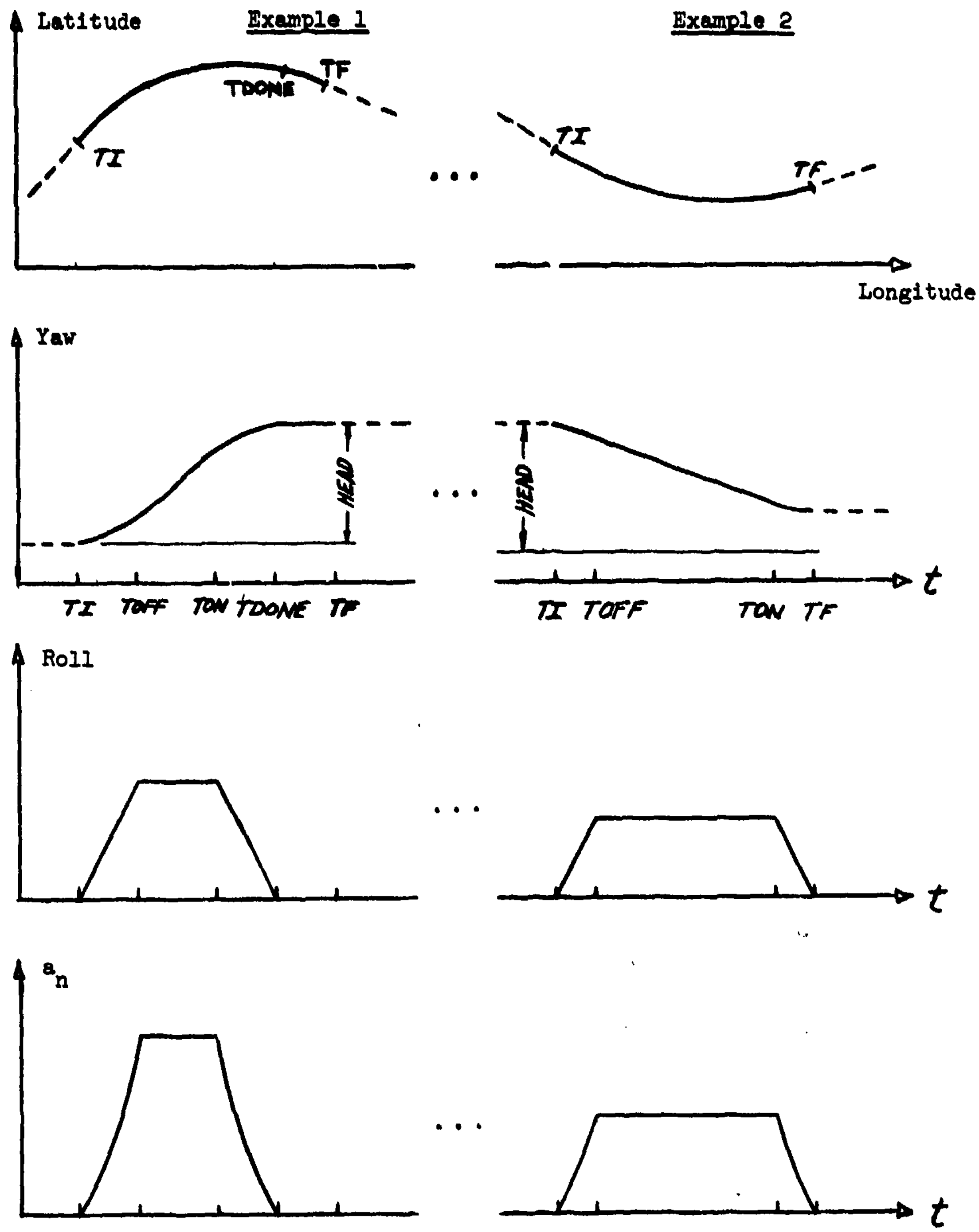


Figure 6 - Two Examples of Constant-Speed Horizontal Turns

It is apparent from Figure 5 that roll rate has from one to three points of discontinuity within the segment - one in Case D, two in B and C and three in A. Again, the numerical integration problem that this presents is handled by piecewise integration as explained in Section 3.4.1.

Before integration begins, time points TOFF, TON and TDONE (defined in Fig. 5) are computed in subroutine TSETUP2. The condition on TDONE is that heading at TDONE should be different from heading at TI by HEAD degrees. To compute TDONE, TSETUP2 must account for variations in both acceleration ($a_n(t)$) and speed. The exact equations for doing this are very non-linear and have been approximated in PROFGEN as quadratics in TDONE. If TSETUP2 finds TDONE is larger than TF it makes TDONE equal to TF to keep the turn within the time limit of the segment. Once TDONE is known, TOFF and TON are easily computed based on max roll angle and ROLRATE. As in the vertical turn, PROFGEN assumes that speed remains positive throughout the turn segment, a condition that the user must guarantee.

The following equation is an approximate expression for the time required to complete a turn through $\Delta\psi$ radians.

$$\Delta t = \begin{cases} \dfrac{V_0\,\Delta\psi}{a_n}\cos\eta_y + 2\,(TOFF - TI) & , \dot{V}_0 = 0 \\[2ex] \dfrac{V_0}{\dot{V}_0}\left(\exp\left(\dfrac{\dot{V}_0\,\Delta\psi}{a_n}\cos\eta_y\right) - 1\right) + 2\,(TOFF - TI) & , \dot{V}_0 \neq 0 \end{cases} \qquad (4)$$

where

Δt = time required to turn $\Delta\psi$ radians (>o)

V_o = total speed at TI (>o)

$\Delta\psi$ = turn angle = |HEAD| (>o)

a_n = normal turning acceleration = TACC (>o)

$\dot{V}_o$ = tangential acceleration = PACC

2(TOFF-TI) = time required to roll into and out of turn

$$= 2 \tan^{-1} \left\{ \frac{a_n}{32.2 \cos(n_y)} \right\} \Big/ \text{ROLRATE}$$

This equation is approximately correct for a turn that rolls quickly to its maximum bank angle, holds that angle for awhile and then rolls quickly back to zero (Case A in Figure 5). The error in this equation grows large as $\Delta\psi$ and ROLRATE grow smaller and as PACC and TACC grow larger.

3.4.3 Sine Maneuver

In a sine maneuver the aircraft follows a ground path like that of Figure 7a. This path results when ground heading, $\psi(t)$, is controlled by the equation

$$\psi(t) = \begin{cases} +A \sin^2 \omega t & , \quad 0 \le t < T_p/2 \\ -A \sin^2 \omega t & , \quad T_p/2 \le t < T_p = 2\pi/\omega \end{cases} \qquad (5)$$

where A is maximum heading variation (HEAD) and w is oscillation frequency (PITCH).

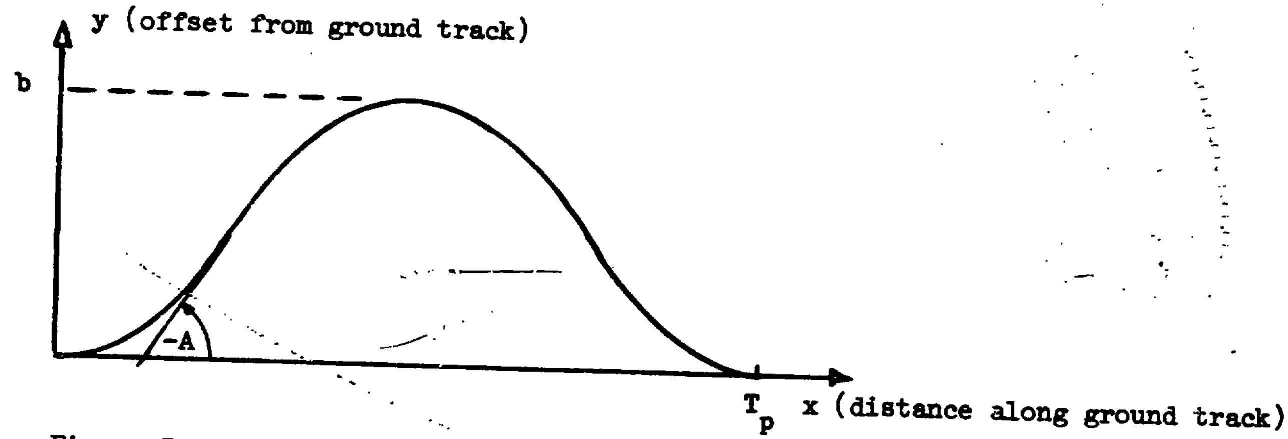


Figure 7a

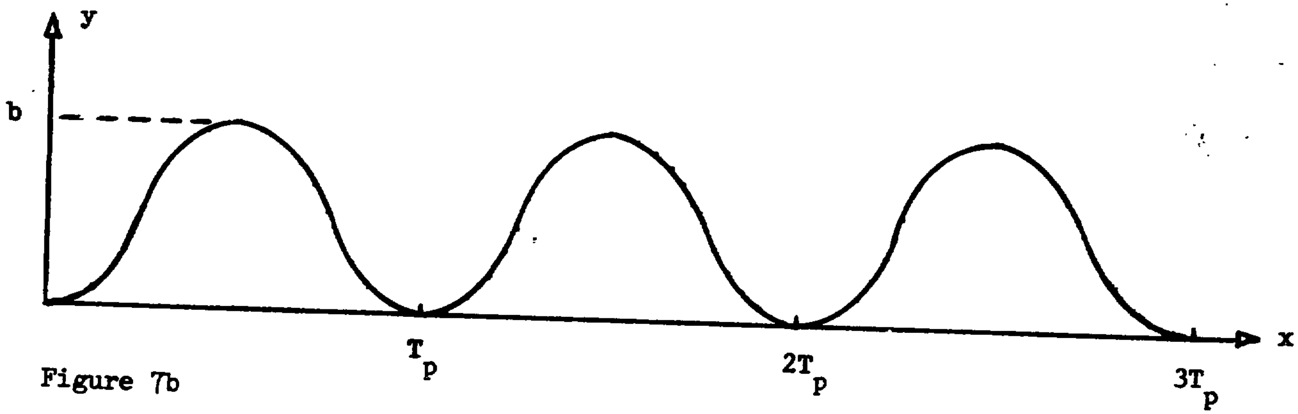


Figure 7b

Figure 7 - Sine Maneuver Ground Tracks

Repeated cyles of 7a are shown in 7b and are produced by simply iterating the above equation to yield a longer maneuver similar to jinking. Note that neither 7a or 7b are properly scaled.

A sine maneuver may execute in any pitch attitude except ±90 degrees and is always performed in coordinated fashion. Again, heading and roll must both be controlled but the governing equation is the one for heading given above. The companion equation for roll that produces coordinated maneuvers is

$$\eta_x = \tan^{-1}\left\{\frac{VA\omega}{32.2}\sin(2\omega t)\right\} \qquad (6)$$

where V is total speed. Since $\dot{\eta}_x$ has no discontinuities, the numerical integration can proceed uninterrupted and the sine maneuver thereby avoids complex event-time calculations like those for a horizontal turn.

Figure 8 shows ground track, roll and heading curves (to scale) for a sine maneuver where A is -20°, T_p is 10 seconds, V is 1000 fps and SEGLNT is 12.5 seconds. Note that roll passes through zero at multiples of $T_p/4$ seconds so that the aircrafts wings are level when the segment is finished at 12.5 seconds.

3.4.4 Straight Flight

Complete straight-flight segments occur when TURN is 4 and partial segments occur anytime a vertical or horizontal turn has reached its max turn angle with time remaining in the segment. Neither roll nor

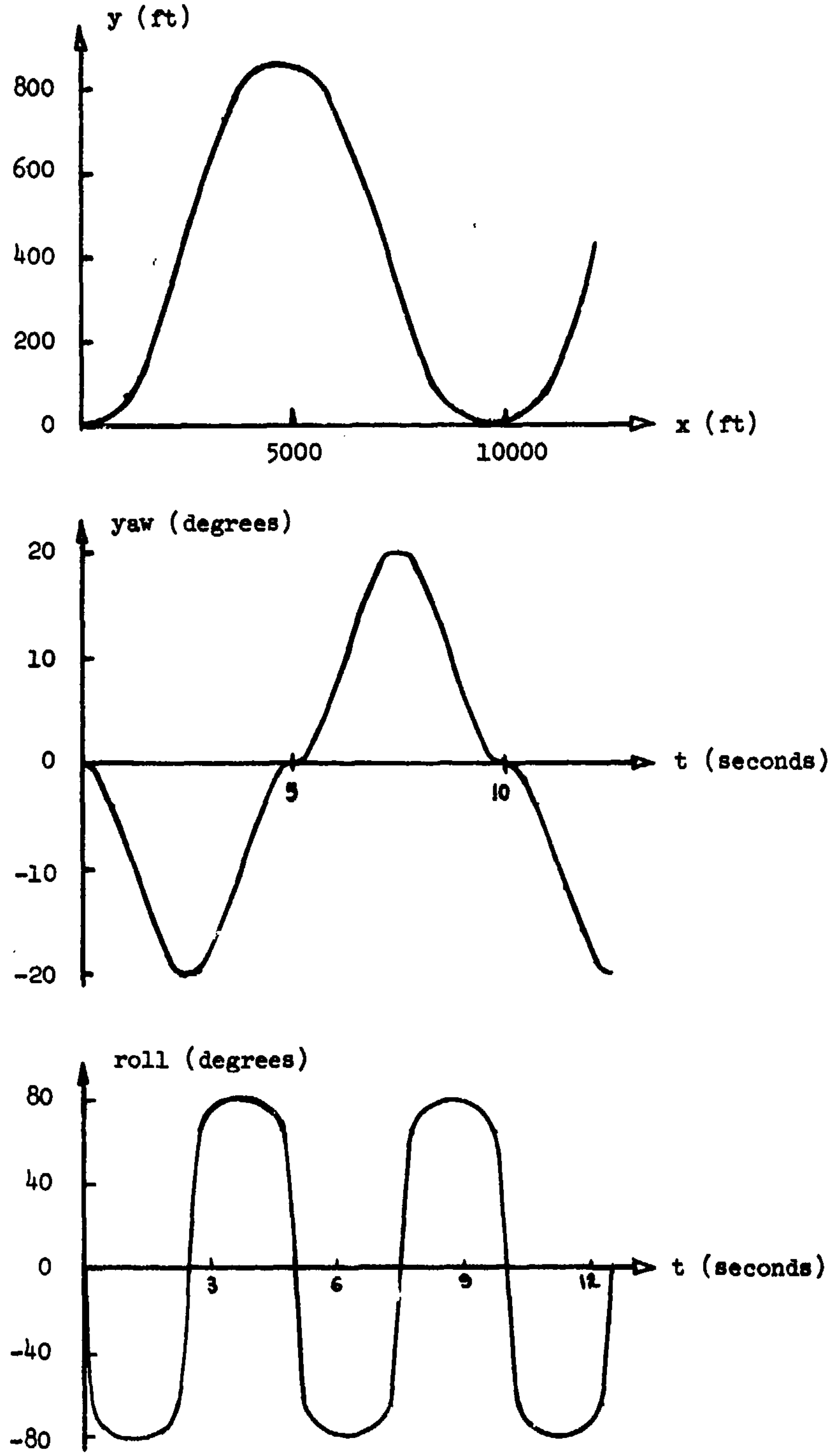


Figure 8 - Example of Sine Maneuver

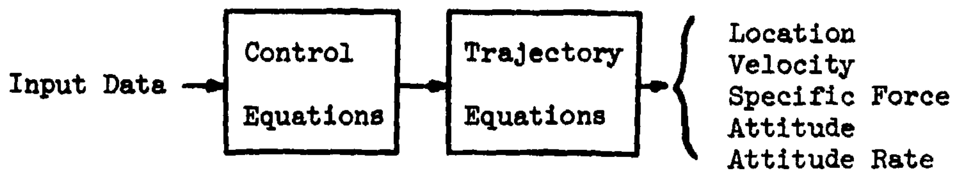
pitch vary in straight flight segments and heading is governed by the users choice of nominal path (NPATH). Heading is constant over a rhumb line path whereas, for a great circle path, heading must vary to keep the aircraft in the great circle plane. Rhumb line flights that continue long enough spiral in on one of the earth's poles and end up causing a division-by-zero failure.

Total speed, which had to remain positive during turning maneuvers, may be zero in straight flight segments. At such times, aircraft position is fixed and attitude is that which existed just prior to speed becoming zero.

To aid the user in constructing straight flight segments between locations over the earth, a program called HEADING has been written. In response to user inputs of lat, lon and altitude at origin and destination, HEADING computes the heading angle at origin needed to reach destination over a great circle path. HEADING also computes the great circle distance from origin to destination. HEADING is a double precision FORTRAN program that can be made available to interested users.

IV. ANALYTICAL DEVELOPMENT

This section develops the equations that govern the trajectory of an aircraft under continuous control in the earth's gravity field. These equations can be conveniently divided into two groups, control equations and trajectory equations, which are related schematically as follows:



The control equations are the relationships that specify turn rates according to the user's input data.

The trajectory equations are a collection of differential and algebraic equations that produce position, velocity, specific force, attitude and attitude rate in response to the imposed control. They are, in short, the equations of motion for a body free to move in six directions in inertial space.

The trajectory equations are kinematic relationships, i.e. they deal with motion in the abstract without reference to force or mass. Since force/mass concepts are immaterial, PROFGEN avoids all aircraft-specific considerations such as moment of inertia, aerodynamic force and thrust force. It follows that the aircraft modeled here is a weightless body that can be displaced and rotated, without restriction, to suit the users demands.

In the following development those equations that became part of the actual code in PROFGEN have stars (*) beside their numbers.

4.1 Coordinate System Descriptions and Relationships

The coordinate systems of particular interest in this report are the inertial, earth, navigation and path systems. These four systems, or frames, will be defined shortly as right-handed orthogonal frames. The relationship of the earth and navigation frames will determine aircraft location (longitude, latitude, alpha) while that of the navigation and path frames will determine attitude (roll, pitch, yaw). Location and attitude data will be carried in two direction cosine matrices (C_e^n and C_p^n) that describe the rotations between pairs of coordinate frames. The subsequent portions of this section describe the four frames, define the two direction cosine matrices and delineate the extraction of location and attitude angles from each of these matrices.

4.1.1 Frame Descriptions

- Inertial frame (i frame: X_i, Y_i, Z_i axes)

  The inertial frame has its origin at the earth's center of mass and is non-rotating relative to the stars. This frame is important mainly as it applies to the computation of specific force. Its relationship to the earth frame is portrayed in Figure 9.

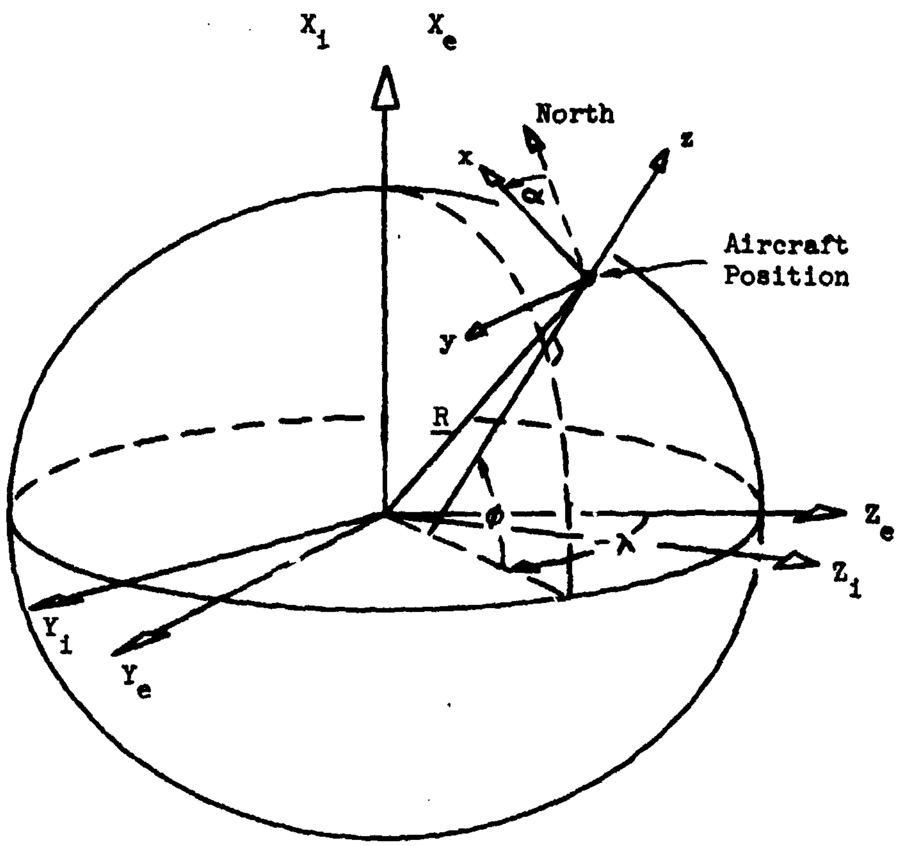


Figure 9 - Earth, Inertial and Navigation Coordinate Frames

- Earth frame (e frame: X_e, Y_e, Z_e axes)

  The earth frame has its origin at the earth's center of mass and has axes fixed in the earth, Figure 9. Axes Y_e, Y_i, Z_e and Z_i all lie in the earth's equatorial plane while axes X_e and X_i are coincident, passing through both poles. The rate of rotation between these two frames is the earth sidereal rate, designated Ω. WGS-72 (Reference 1) gives this value for Ω which is denoted WEI in PROFGEN:

$$\Omega = 0.7292115147 \times 10^{-4} \text{ rad/sec}$$

- Navigation frame (n frame: x, y, z axes)

  This locally-level frame has its origin at the aircraft center of mass with x and y in a plane tangent to the reference ellipsoid and z perpendicular to the ellipsoid, Figure 9. (Center of mass and center of rotation are coincident in this development). PROFGEN solves the trajectory equations in the navigation frame. Aircraft location is specified relative to the earth frame by the three-tuple (λ, ϕ, α) where λ is longitude, ϕ is geographic latitude and α is the navigation frame heading angle, referred to variously as alpha, wander angle or wander azimuth angle. Figure 9 shows that ϕ is geographic latitude, not geocentric latitude. Thus z is normal to the elliptical

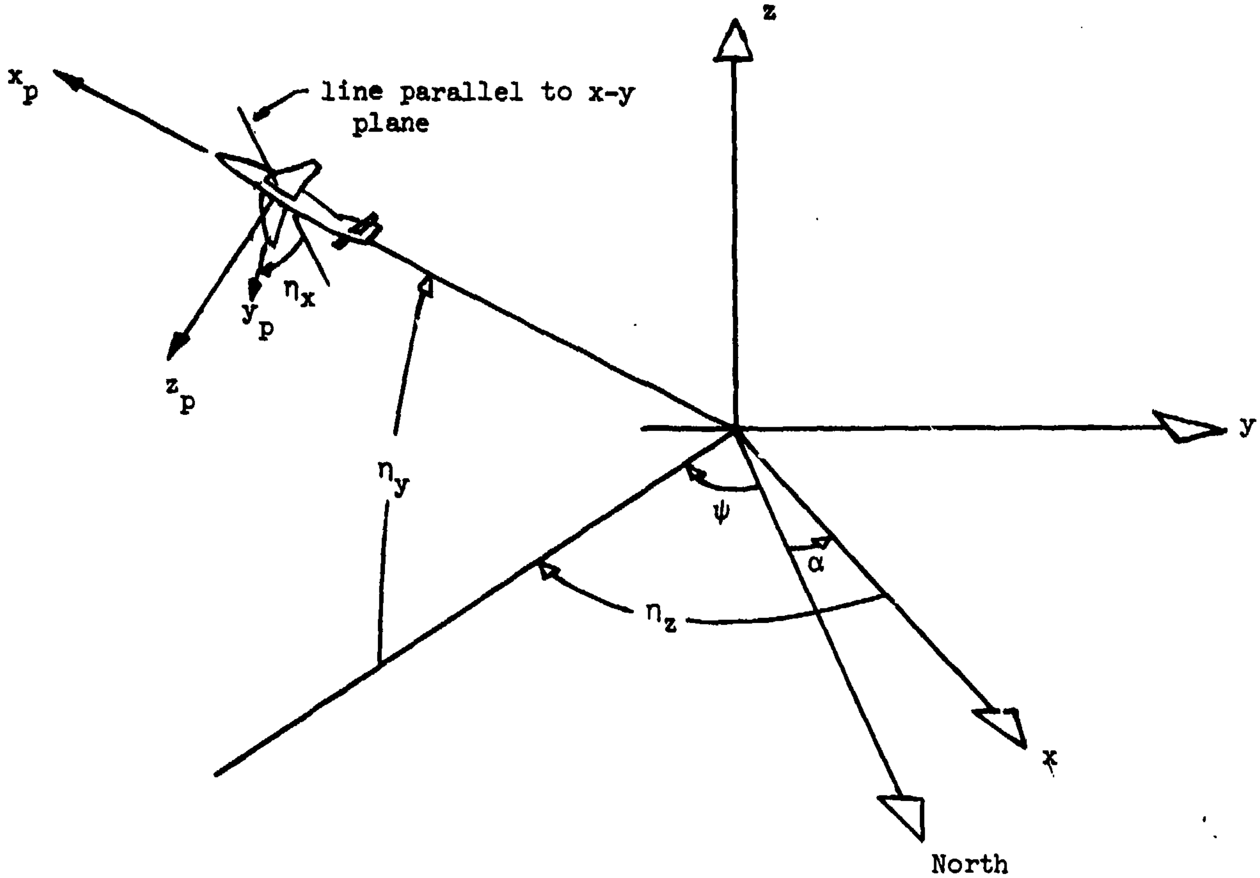
surface of the earth rather than in the direction of the earth center. The values for λ, ϕ, and α will be computed from the direction cosine matrix C_e^n.

- Path frame (p frame: x_p, y_p, z_p axes)

The path frame, depicted in Figure 10, has its origin at the aircraft center of mass. It takes its name from the fact that the x_p-axis follows the aircraft path by staying aligned with the total velocity vector, $\underline{V}$. ($\underline{V}$, velocity with respect to the earth, will be defined precisely in Section 4.2.3)

In general $\underline{V}$ is misaligned from the aircraft's longitudinal axis by an angle of attack and a crab angle. In this development we assume these angles are zero. The effect of this assumption is to weld the path frame to the aircraft's body thus causing x_p to pass through the aircraft nose and y_p to point out the right wing. z_p points down in level flight but rotates about x_p during coordinated turns so there is never any maneuver acceleration along y_p.

Since path and body are coincident, the familiar body frame terms of roll, pitch and yaw will be borrowed to describe the Euler angles between the path and navigation frames. Roll, pitch and yaw are denoted η_x, η_y and η_z and are



Note: Origin of path frame displaced from that of nav frame only for clarity of diagram; they are actually coincident at aircraft center of mass.

Figure 10 - Navigation and Path Coordinate Frames

measured around x_p, y_p and z_p respectively. A right turn produces a positive yaw rotation, a pitch up is a positive pitch rotation, and a clockwise roll (as viewed from behind the aircraft) is a positive roll rotation. The values of η_x, η_y and η_z will be computed from the direction cosine matrix C_p^n.

4.1.2 Frame Relationships: Direction Cosines and Euler Angles

- Earth and Navigation Frames

Figure 9 presents the relationship between the earth and navigation frames. When λ, ϕ and α are zero, the navigation frame is directionally aligned with the earth frame. Beginning at the aligned position, the rotations necessary to go from earth to nav coordinates form the direction cosine matrix C_e^n. This matrix is the ordered product of three individual matrices describing these rotations: an x rotation of λ degrees, a y rotation of ϕ degrees and a z rotation of α degrees. Using an "s" prefix for the trigonometric sine and a "c" prefix for the cosine, C_e^n is

$$C_e^n = \begin{bmatrix} c\alpha & s\alpha & 0 \\ -s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\phi & 0 & -s\phi \\ 0 & 1 & 0 \\ s\phi & 0 & c\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\lambda & s\lambda \\ 0 & -s\lambda & c\lambda \end{bmatrix} \quad (7)$$

Now if the elements of C_e^n are identified as

$$C_e^n = \begin{bmatrix} CEN_{11} & CEN_{12} & CEN_{13} \\ CEN_{21} & CEN_{22} & CEN_{23} \\ CEN_{31} & CEN_{32} & CEN_{33} \end{bmatrix}$$

then the individual elements are

$$\begin{aligned}
CEN_{11} &= \cos\alpha \cos\phi \\
CEN_{21} &= -\sin\alpha \cos\phi \\
CEN_{31} &= \sin\phi \\
CEN_{12} &= \sin\alpha \cos\lambda + \cos\alpha \sin\phi \sin\lambda \\
CEN_{22} &= \cos\alpha \cos\lambda - \sin\alpha \sin\phi \sin\lambda \\
CEN_{32} &= -\cos\phi \sin\lambda \\
CEN_{13} &= \sin\alpha \sin\lambda - \cos\alpha \sin\phi \cos\lambda \\
CEN_{23} &= \cos\alpha \sin\lambda + \sin\alpha \sin\phi \cos\lambda \\
CEN_{33} &= \cos\phi \cos\lambda
\end{aligned} \qquad (9)^*$$

* Coded for implementation in PROFGEN.

To extract latitude, longitude and alpha from the elements of C_e^n, the following calculations are made

$$\phi = \sin^{-1}(CEN_{31}) \quad , \phi \in [-\pi/2, +\pi/2] \qquad (10)^*$$

$$\lambda = \tan^{-1}(-CEN_{32}/CEN_{33}) , \lambda \in [-\pi, +\pi] \qquad (11)^*$$

$$\alpha = \tan^{-1}(-CEN_{21}/CEN_{11}) \quad , \alpha \in [-\pi, +\pi] \qquad (12)^*$$

where the initial values for ϕ, λ and α are

$$\phi = \text{LATO}$$

$$\lambda = \text{LONO}$$

$$\alpha = \text{ALFAO}$$

The FORTRAN functions SIN ($\cdot$) and ATAN2 ($\cdot$, $\cdot$) were used to implement (10), (11) and (12) because their range agrees with that desired for ϕ, λ and α. An important aspect of the computation for λ in Equation (11) is that $\phi \in [-\pi/2, \pi/2]$, which means $\cos(\phi)$ is always positive, which in turn makes the sign of CEN_{32} and CEN_{33} depend solely on λ, which removes any doubt as to the quadrant where λ lies. A similar statement applies to α as computed in (12).

● Path and Navigation Frames

Figure 10 presents the relationship between the path and navigation frames. Beginning at the nonaligned position shown there, the ordered sequence of rotations necessary to form the C_p^n matrix is as follows: a roll about x_p of η_x degrees to get the wings level; a pitch about y_p of η_y degrees to get the nose level; a yaw about z_p of η_z degrees to align the x_p and x axes; finally, a flip about x_p of 180° to align z_p, which is nominally down, with z which is always up. Thus

$$C_p^n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} c\eta_z & -s\eta_z & 0 \\ s\eta_z & c\eta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\eta_y & 0 & s\eta_y \\ 0 & 1 & 0 \\ -s\eta_y & 0 & c\eta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\eta_x & -s\eta_x \\ 0 & s\eta_x & c\eta_x \end{bmatrix}$$

Now if the elements of C_p^n are identified as

$$C_p^n = \begin{bmatrix} CPN_{11} & CPN_{12} & CPN_{13} \\ CPN_{21} & CPN_{22} & CPN_{23} \\ CPN_{31} & CPN_{32} & CPN_{33} \end{bmatrix} \tag{14}$$

then the individual elements are

$$CPN_{11} = \cos\eta_z \cdot \cos\eta_y$$

$$CPN_{21} = -\sin\eta_z \cdot \cos\eta_y$$

$$CPN_{31} = \sin\eta_y$$

$$CPN_{12} = \cos\eta_z \cdot \sin\eta_y \cdot \sin\eta_x - \sin\eta_z \cdot \cos\eta_x$$

$$CPN_{22} = -\sin\eta_z \cdot \sin\eta_y \cdot \sin\eta_x - \cos\eta_z \cdot \cos\eta_x \qquad (15)^*$$

$$CPN_{32} = -\cos\eta_y \cdot \sin\eta_x$$

$$CPN_{13} = \cos\eta_z \cdot \sin\eta_y \cdot \cos\eta_x + \sin\eta_z \cdot \sin\eta_x$$

$$CPN_{23} = \cos\eta_z \cdot \sin\eta_x - \sin\eta_z \cdot \sin\eta_y \cdot \cos\eta_x$$

$$CPN_{33} = -\cos\eta_y \cdot \cos\eta_x$$

Roll, pitch and yaw are extracted from the elements of C_p^n as follows:

$$\eta_x = \tan^{-1}(-CPN_{32} / -CPN_{33}) \quad , \; \eta_x \in [-\pi, +\pi] \qquad (16)^*$$

$$\eta_y = \sin^{-1}(CPN_{31}) \quad , \; \eta_y \in [-\pi/2, +\pi/2] \qquad (17)^*$$

$$\eta_z = \tan^{-1}(-CPN_{21} / CPN_{11}) \quad , \; \eta_z \in [-\pi, +\pi] \qquad (18)^*$$

where the initial values are

$$\eta_x = 0$$

$$\eta_y = \text{PPITCHO}$$

$$\eta_z = \text{ALFAO} + \text{PHEADO}$$

Again SIN (·) and ATAN2 (· , ·) were used to implement (16), (17) and (18). As with λ and α, the key to the computations in (16) and (18) lies in the fact that η_y has a restricted range which makes its cosine always positive. The relationship between α, η_z and ψ (heading) is illustrated in Figure 11.

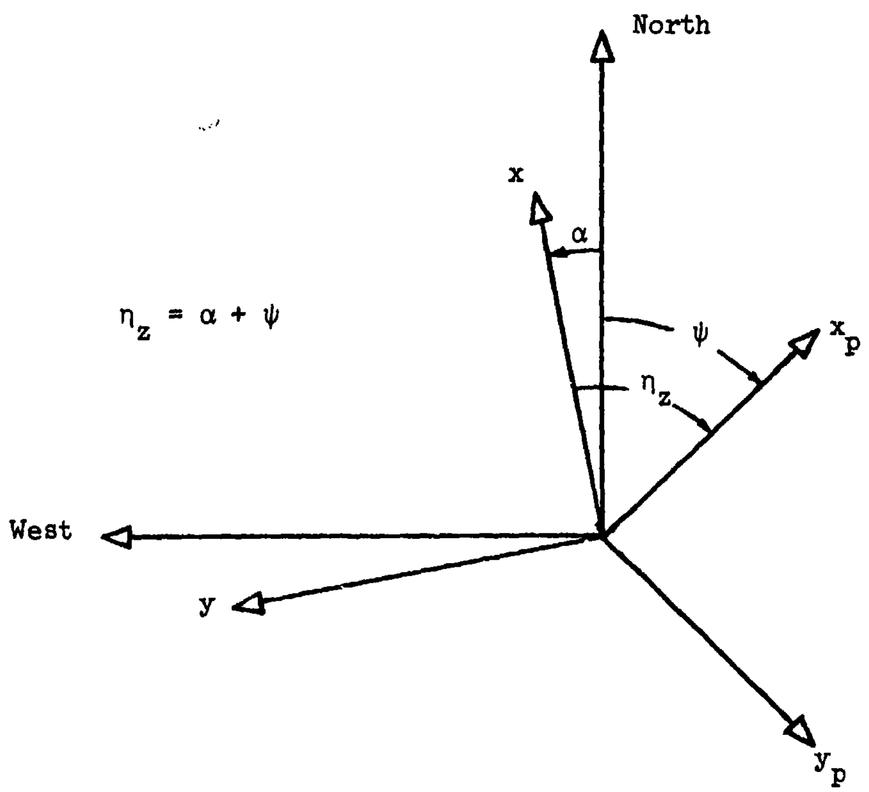


Figure 11 - Relationship of η_z, α and ψ

4.2 Trajectory Equations

Sections 4.2.1, 4.2.2 and 4.2.3 will develop first order differential equations to describe the motion of a body in six degrees of freedom. Section 4.2.4 defines the states of the state vector, $\underline{x}$. The companion algebraic relationships for specific force, attitude rates and plumb-bob gravity will be developed in Section 4.2.5.

4.2.1 Direction Cosine Rates: Location and Attitude

At least three methods are available for keeping track of the rotation angles between frames, including direct integration of the Euler angle rates, propagation of four quaterion parameters representing a complete direction cosine matrix (Reference 5), and propagation of the direction cosine matrix. The last approach was chosen for PROFGEN because of its simplicity and versatility. This section derives a general expression for the direction cosine rate and then displays the result in notation appropriate to C_e^n and C_p^n.

For any two frames, a and b, the Theorm of Coriolis can be written for any vector $\underline{u}$ as

$$\left.\frac{d\underline{u}}{dt}\right|_a = \left.\frac{d\underline{u}}{dt}\right|_b + \underline{\beta}_{ba} \times \underline{u} \qquad (1)$$

This equation is in "physical vector" form. It states that the time rate of change of $\underline{u}$, as observed in the a frame (i.e. with respect to the a frame), equals the time rate of change of $\underline{u}$, as observed in the b frame, plus the angular rate of change of frame b with respect to frame a crossed onto $\underline{u}$. The addition and multiplication in (19) are physical-vector addition and physical-vector cross multiplication. When (19) is coordinatized in the a frame, these "math vector" relationships follow:

$$\left(\frac{d\underline{u}}{dt}\bigg|_a\right)^a = \left(\frac{d\underline{u}}{dt}\bigg|_b + \underline{\beta}_{ba} \times \underline{u}\right)^a$$

$$= \left(\frac{d\underline{u}}{dt}\bigg|_b\right)^a + B_{ba}^a \, \underline{u}^a$$

$$= C_b^a \left(\frac{d\underline{u}}{dt}\bigg|_b\right)^b + B_{ba}^a \, C_b^a \, \underline{u}^b$$

or

$$\dot{\underline{u}}^a = C_b^a \, \dot{\underline{u}}^b + B_{ba}^a \, C_b^a \, \underline{u}^b \qquad (20)$$

where B^a_{ba} is a "cross-matrix" that produces a result on a math vector identical to that of cross multiplication on a physical vector. B^a_{ab} is defined below. Continuing

$$\underline{\underline{u}}^a = C^a_b \underline{\underline{u}}^b$$

$$\dot{\underline{\underline{u}}}^a \triangleq \frac{d}{dt} \underline{\underline{u}}^a = \frac{d}{dt}\left(C^a_b \underline{\underline{u}}^b\right)$$

$$= C^a_b \dot{\underline{\underline{u}}}^b + \dot{C}^a_b \underline{\underline{u}}^b \qquad (21)$$

Equating (20) and (21) yields

$$\dot{C}^a_b \underline{\underline{u}}^b = B^a_{ba} C^a_b \underline{\underline{u}}^b$$

and, since u is any vector, it follows that

$$\dot{C}^a_b = B^a_{ba} C^a_b \qquad (22)$$

where

$$B_{ba}^{a} = \begin{bmatrix} 0 & -\beta_z & \beta_y \\ \beta_z & 0 & -\beta_x \\ -\beta_y & \beta_x & 0 \end{bmatrix} \tag{23}$$

$$\begin{pmatrix} \beta_x \\ \beta_y \\ \beta_z \end{pmatrix} = \underset{\sim}{\beta}_{ba}^{a} \tag{24}$$

The specific notation chosen to implement (22) and (24) for C_e^n and C_p^n is shown below:

$$\dot{C}_e^n = -\begin{bmatrix} 0 & -\rho_z & \rho_y \\ \rho_z & 0 & -\rho_x \\ -\rho_y & \rho_x & 0 \end{bmatrix} C_e^n \tag{25}^*$$

where

$$\begin{pmatrix} \rho_x \\ \rho_y \\ \rho_z \end{pmatrix} \triangleq \underset{\sim}{\rho}_{ne}^{n} = -\underset{\sim}{\rho}_{en}^{n} \tag{26}^*$$

Also

$$\dot{C}_p^n = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} C_p^n \qquad (27)^*$$

where

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \triangleq \underline{\omega}_{pn}^n \qquad (28)^*$$

For writing convenience, $\underline{\rho}_{ne}^n$ and $\underline{\omega}_{pn}^n$ will be referred to hereafter as $\underline{\rho}$ and $\underline{\omega}$. In (25) and (27) we have expressions for keeping track of location and attitude provided $\underline{\rho}$ and $\underline{\omega}$ can be computed. Sections 4.2.2 and 4.2.3 deal with $\underline{\rho}$. The computation for $\underline{\omega}$ will be given in Section 4.3 where turning rates are discussed.

4.2.2 Angular Rate - Nav Frame w.r.t. Earth Frame

$\underline{\rho}$ is the angular rate of the nav frame with respect to the earth frame. The fact that the x and y axes of the nav frame remain tangent to the earth will be used to derive expressions for ρ_x and ρ_y. ρ_z will be determined by the users choice of azimuth-angle mechanization.

Consider the geometry of Figure 12, a section of the earth ellipsoid, where V_N and V_W denote North and West velocity components. The North-West-Up (N-W-U) frame differs from the nav frame only by the rotation α. If $\underline{V}$ is earth frame velocity, its navigation frame components are denoted

$$\underline{V}^n \triangleq \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} \qquad (29)$$

Then from Figure 11, V_N, V_W, V_{UP} are given by

$$\begin{pmatrix} V_N \\ V_W \\ V_{UP} \end{pmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} \qquad (30)^*$$

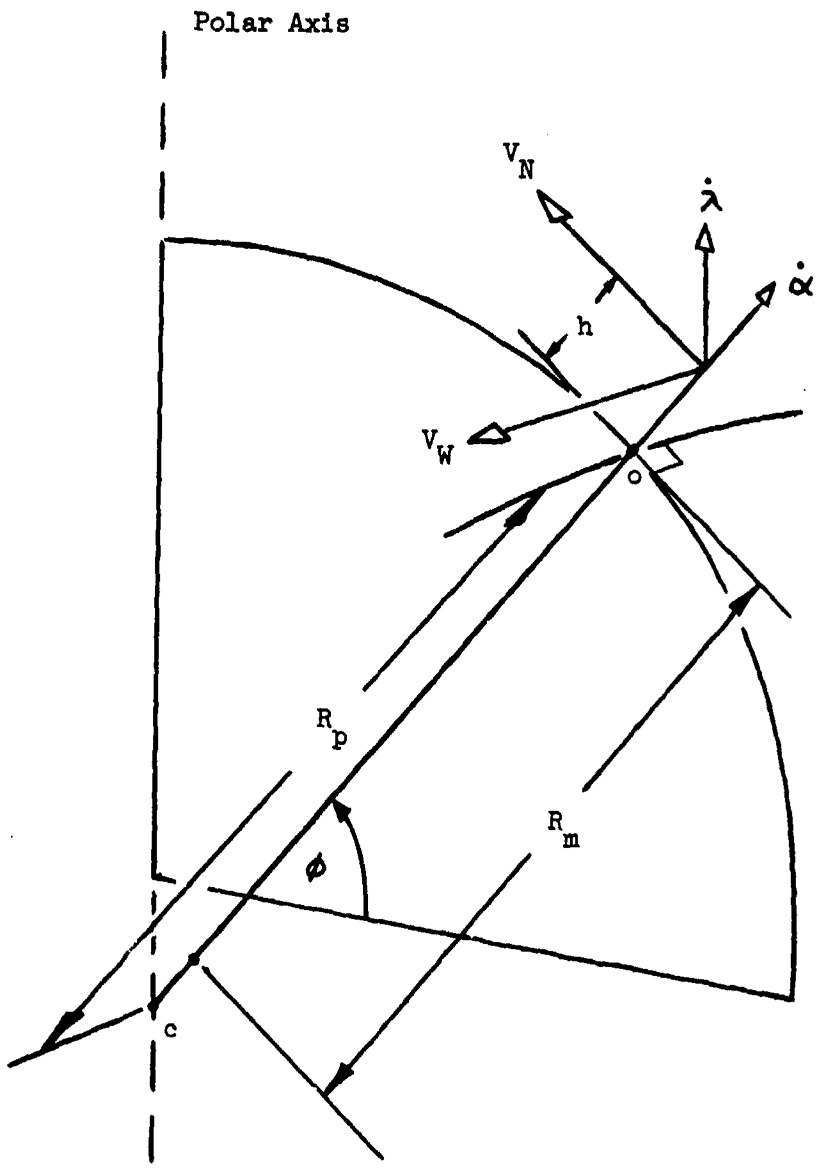


Figure 12 - Geometry for Deriving ρ

The angular rates required to keep the N-W-U frame level over the earth ellipsoid are deduced from Figure 12 as

$$\rho_N = \frac{-V_W}{R_p + h} \qquad (31)^*$$

$$\rho_W = \frac{V_N}{R_m + h} \qquad (32)^*$$

where h is altitude above the ellipsoid, R_m is the radius of curvature of an ellipsoid meridian line and R_p is the radius of curvature of the ellipsoid in a plane through the normal and at right angles to the meridian. (It can be shown that R_p is the distance $\overline{oc}$ where c lies on the polar axis.) R_m and R_p vary with ϕ according to the following equations (Ref. 2, pp 168-170):

$$R_m = \frac{R_e (1 - e^2)}{(1 - e^2 \sin^2\phi)^{3/2}} \qquad (33)^*$$

$$R_p = \frac{R_e}{(1 - e^2 \sin^2\phi)^{1/2}} \qquad (34)^*$$

where

e^2 = eccentricity^2 = $\frac{R_e^2-b^2}{R_e^2}$ = 0.006694317778 (WGS-72 data)

R_e = semimajor earth axis = 20925640 feet (WGS-72)

b = semiminar earth axis = 20855481 feet (WGS-72)

ρ_N and ρ_W lie in the x-y plane of the nav frame and can be resolved into components along x and y as follows:

$$\begin{aligned} \rho_x &= \rho_N \cos\alpha + \rho_W \sin\alpha \\ \rho_y &= -\rho_N \sin\alpha + \rho_W \cos\alpha \\ \rho_z &= \rho_{UP} \end{aligned} \qquad (35)^*$$

The general relationship between ρ_z and α can be deduced from the geometry of Figure 12 as

$$\rho_z = \dot{\lambda} \sin\phi + \dot{\alpha} \qquad (36)^*$$

The value for $\dot{\alpha}$ depends on the azimuth angle mechanization (LLMECH) desired by the user. The various choices and the resulting ρ_z values are tabulated in Table 2.

LLMECH	Name	$\dot{\alpha}$	ρ_z
1	Alpha Wander	$-\dot{\lambda} \sin \phi$	0
2	Constant Alpha	0	$\dot{\lambda} \sin \phi$
3	Unipolar	$-J \dot{\lambda}$ †	$\dot{\lambda} (\sin \phi - J)$
4	Free Azimuth	$-(\Omega + \dot{\lambda}) \sin \phi$ ††	$-\Omega \sin \phi$

† $J \triangleq \text{sign}(\phi)$ †† $\Omega = 0.7292115147 \times 10^{-4}$ rad/sec

Table 2 - Azimuth Angle Mechanization Schemes

Figure 13, a section of the earth, is drawn so V_W is perpendicular to the paper at the indicated point. (Note again that R_p terminates on the polar axis.) Examination of this figure shows that the equation for $\dot{\lambda}$ is

$$\dot{\lambda} = \frac{-V_W}{(R_p + h) \cos \phi} \qquad (37)^*$$

ρ_x, ρ_y and ρ_z, as derived in this section, depend on α, ϕ, V_x, V_y and V_z. α and ϕ can be obtained from C_e^n using Equations (10) and (12) while expressions for V_x, V_y and V_z will be derived in the next section.

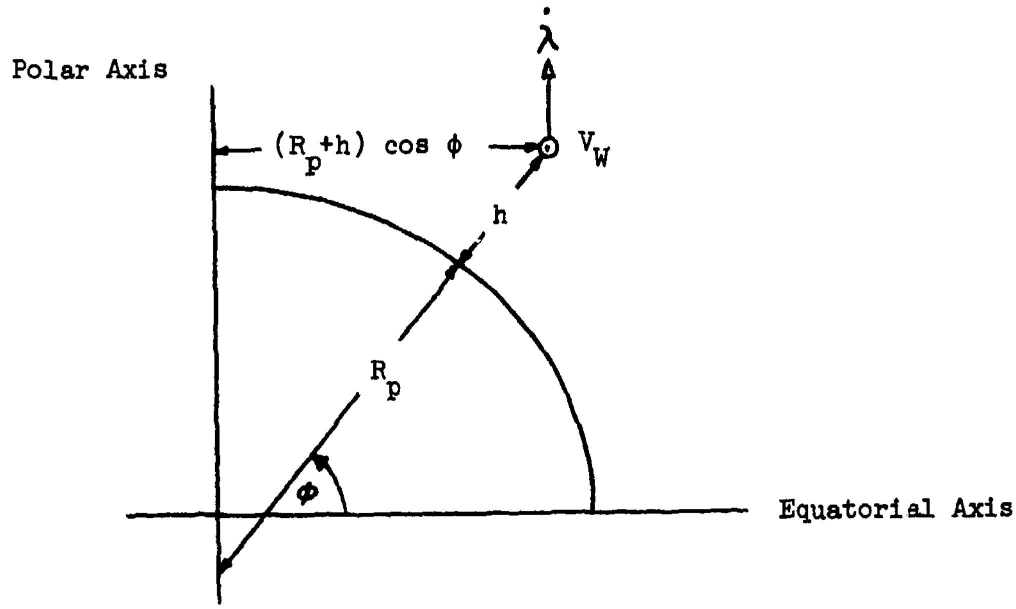


Figure 13 - Geometry for Deriving $\dot{\lambda}$

4.2.3 Velocity w.r.t. Earth

Referring now to Figure 9, the vector $\underline{R}$ connects the earth's center with the aircraft location at all times. By definition of $\underline{V}$ and by Coriolis' Law

$$\underline{V} \triangleq \left.\frac{d\underline{R}}{dt}\right|_e \qquad (38)$$

$$\left.\frac{d\underline{V}}{dt}\right|_n = \left.\frac{d\underline{V}}{dt}\right|_p + \underline{\omega}_{pn} \times \underline{V} \qquad (39)$$

Coordinatize (39) in the nav frame and use (22) and (28) to produce the following equivalent expressions in math-vector form:

$$\left(\frac{d\underline{V}}{dt}\Big|_n\right)^n = \left(\frac{d\underline{V}}{dt}\Big|_p\right)^n + \Omega_{pn}^n\,\underline{V}^n$$

$$\triangleq \left(\frac{d\underline{V}}{dt}\Big|_p\right)^n + \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}$$

$$= C_p^n\left(\frac{d\underline{V}}{dt}\Big|_p\right)^p + \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} \tag{40}$$

Since velocity in the path frame lies entirely along the x_p axis

$$\underline{V}_p = \begin{pmatrix} \sqrt{V_x^2 + V_y^2 + V_z^2} \\ 0 \\ 0 \end{pmatrix} \triangleq \begin{pmatrix} V_T \\ 0 \\ 0 \end{pmatrix} \tag{41}$$

$$\left(\frac{d\underline{V}}{dt}\Big|_p\right)^p = \begin{pmatrix} \dot{V}_T \\ 0 \\ 0 \end{pmatrix} \tag{42}$$

Substituting (42) in (40) and writing out the individual equations yields

$$\dot{V}_x = CPN_{11}\,\dot{V}_T - \omega_z V_y + \omega_y V_z$$
$$\dot{V}_y = CPN_{21}\,\dot{V}_T + \omega_z V_x - \omega_x V_z \qquad (43)^*$$
$$\dot{V}_z = CPN_{31}\,\dot{V}_T - \omega_y V_x + \omega_x V_y$$

$\dot{V}_T$ will be recognized as PACC, the path acceleration needed to alter the magnitude of $\underline{V}$.

$$\dot{V}_T = PACC \qquad ft/sec^2 \qquad (44)^*$$

In (43) we have a differential equation for earth frame velocity that depends only on factors already specified save for $\underline{\omega} = (\omega_x\ \omega_y\ \omega_z)^T$. To repeat, $\underline{\omega}$ will be derived in Section 4.3

4.2.4 State Vector

PROFGEN carries a state vector, $\underline{x}$, containing 23 states in a 23 element, labled-common array named STATE:

$$\underline{x} = (V_x\ V_y\ V_z\ V_T\ h\ CPN_{11}\ CPN_{21}\ \cdots\ CPN_{33}\ CEN_{11}\ CEN_{21}\ \cdots\ CEN_{33})^T_{23\times1}$$

The appropriate differential equations for the elements of $\underline{x}$ are Equation (43) for the velocity components V_x, V_y, V_z; Equation (44) for the total velocity V_T; Equation (25) for attitude data in C_p^n; Equation (27) for the location data in C_e^n and this differential equation for altitude, h;

$$\dot{h} = V_z \qquad (45)^*$$

4.2.5 Other Trajectory Relationships

The following three topics are discussed now to conclude the derivation of the trajectory equations:

a. Specific Force

b. Attitude Rates

c. Gravity Model

Topic c supports topic a. Topics a and b are important only insofar as they provide a way to compute specific force and attitude rate for PROFGEN output. Specific force and attitude rate are algebraic expressions not required during state vector propagation; therefore, in some sense, these equations lie outside the mainstream of PROFGEN's calculations.

a. Specific Force

Specific force, $\underline{F}$, is the acceleration that a velocity meter (accelerometer) aboard the aircraft would detect. Specific force is the total inertial acceleration minus the mass-attraction gravitational acceleration; i.e. specific force is the second rate of change of $\underline{R}$ as viewed by an observer fixed in inertial space, minus mass-attraction gravity, $\underline{G}_m$. The physical vector equation for this (see Reference 3, p. 121), where + and - are physical vector operations, is

$$\underline{F} = \left.\frac{d^2\underline{R}}{dt^2}\right|_i - \underline{G}_m \tag{46}$$

Recall from (38) that

$$\underline{V} \triangleq \left.\frac{d\underline{R}}{dt}\right|_e \tag{38}$$

The e frame rotates at rate $\underline{\Omega}$ ($\underline{\Omega}^e = (\Omega\ 0\ 0)^T$) with respect to the inertial frame so we can write

$$\left.\frac{d\underline{R}}{dt}\right|_i = \left.\frac{d\underline{R}}{dt}\right|_e + \underline{\Omega} \times \underline{R}$$

$$= \underline{V} + \underline{\Omega} \times \underline{R} \tag{47}$$

The navigation frame rotates at rate $\underline{T}$ ($\underline{T} \triangleq \underline{\rho} + \underline{\Omega}$) with respect to the inertial frame. Differentiating (47), substituting the result in (46), and continuing with the expansion gives

$$\underline{F} = \frac{d}{dt}\left(\underline{V} + \underline{\Omega} \times \underline{R}\right)\Big/_i - \underline{G}_m$$

$$= \frac{d\underline{V}}{dt}\Big/_i + \frac{d}{dt}\left(\underline{\Omega} \times \underline{R}\right)\Big/_i - \underline{G}_m$$

$$= \frac{d\underline{V}}{dt}\Big/_n + \underline{T} \times \underline{V} + \frac{d}{dt}\left(\underline{\Omega} \times \underline{R}\right)\Big/_i - \underline{G}_m$$

$$= \frac{d\underline{V}}{dt}\Big/_n + \underline{T} \times \underline{V} + \frac{d}{dt}\left(\underline{\Omega} \times \underline{R}\right)\Big/_e + \underline{\Omega} \times \left(\underline{\Omega} \times \underline{R}\right) - \underline{G}_m$$

$$= \frac{d\underline{V}}{dt}\Big/_n + \left(\underline{\rho} + \underline{\Omega}\right) \times \underline{V} + \underline{\Omega} \times \underline{V} + \underline{\Omega} \times \left(\underline{\Omega} \times \underline{R}\right) - \underline{G}_m$$

$$= \frac{d\underline{V}}{dt}\Big/_n + \left(\underline{\rho} + 2\underline{\Omega}\right) \times \underline{V} - \underline{g} \qquad (48)$$

where we have used the fact $d\underline{\Omega}/dt|_e = \underline{0}$ and where

$$\underline{g} \triangleq \underline{G}_m - \underline{\Omega} \times (\underline{\Omega} \times \underline{R}) \tag{49}$$

The vector $\underline{g}$ is the usual plumb-bob gravity composed of both mass attraction and earth rotation components. The vector $\underline{g}$ points downward. Recalling from Section 4.2.3 that

$$\left(\frac{d\underline{V}}{dt}\bigg|_n\right)^n = \dot{\underline{V}}^n = \left(\dot{V}_x \ \dot{V}_y \ \dot{V}_z\right)^T$$

we can componentize (48) in the nav frame (subscripts x, y, z) as follows

$$F_x = \dot{V}_x + (\rho_y + 2\Omega_y)V_z - (\rho_z + 2\Omega_z)V_y - g_x$$

$$F_y = \dot{V}_y + (\rho_z + 2\Omega_z)V_x - (\rho_x + 2\Omega_x)V_z - g_y \tag{50}^*$$

$$F_z = \dot{V}_z + (\rho_x + 2\Omega_x)V_y - (\rho_y + 2\Omega_y)V_x - g_z$$

where

$$\begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} = C_e^n \begin{pmatrix} \Omega \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} CEN_{11} \\ CEN_{21} \\ CEN_{31} \end{pmatrix} \Omega \qquad (51)^*$$

$$\Omega = 0.7292115147 \cdot 10^{-4} \text{ rad/sec}$$

Gravity (g_x, g_y, g_z) will be discussed presently. Ω_x, Ω_y and Ω_z are the projections onto the nav frame of the angular velocity between the earth and inertial frames; these quantities are not related to ω_x, ω_y and ω_z except that both share the greek letter "omega", upper and lower case. All other quantities needed for computing (50) have already been discussed.

b. Attitude Rates

This section derives an expression for each Euler angle rate ($\dot{\eta}_x$, $\dot{\eta}_y$, $\dot{\eta}_z$) as a function of the commanded turning rates, ω_x ω_y and ω_z. Recall these formulas from (15) and (27):

$$CPN_{11} = \cos\eta_z \cos\eta_y \qquad (52a)$$

$$CPN_{21} = -\sin\eta_z \cos\eta_y \qquad (52b)$$

$$CPN_{31} = \sin\eta_y \qquad (52c)$$

$$CPN_{33} = -\cos\eta_y \cos\eta_x \qquad (52d)$$

$$\dot{CPN}_{11} = -\omega_z\, CPN_{21} + \omega_y\, CPN_{31} \qquad (52e)$$

$$\dot{CPN}_{31} = -\omega_y\, CPN_{11} + \omega_x\, CPN_{21} \qquad (52f)$$

$$\dot{CPN}_{33} = -\omega_y\, CPN_{13} + \omega_x\, CPN_{23} \qquad (52g)$$

Differentiate (52c) to get

$$\dot{CPN}_{31} = (\cos \eta_y)\, \dot{\eta}_y$$

and equate this to (52f) to get

$$-\omega_y\, CPN_{11} + \omega_x\, CPN_{21} = (\cos \eta_y)\, \dot{\eta}_y$$

$$-\omega_y\, (\cos \eta_z \cos \eta_y) + \omega_x\, (-\sin \eta_z \cos \eta_y) = (\cos \eta_y)\, \dot{\eta}_y$$

Assume now that pitch is not ±90° and cancel cos η_y to yield

$$\dot{\eta}_y = -\omega_y \cos\eta_z - \omega_x \sin\eta_z \qquad (53)^*$$

$$(\cos\eta_y \neq 0)$$

which is the desired expression for $\dot{\eta}_y$. Similar manipulations of (52) produced the following expressions for $\dot{\eta}_x$ and $\dot{\eta}_z$:

$$\dot{\eta}_x = (\omega_x \cos\eta_z - \omega_y \sin\eta_z)/\cos\eta_y \qquad (\cos\eta_y \neq 0) \tag{54*}$$

$$\dot{\eta}_z = -\omega_z + \tan\eta_y(\omega_x \cos\eta_z - \omega_y \sin\eta_z) \qquad (\cos\eta_y \neq 0) \tag{55*}$$

PROFGEN does not attempt to make attitude rate calculations when $\cos\eta_y = 0$. It simply prints a warning message and goes on (see Section 3.3).

c. Gravity Model

Throughout this report the ellipticity of the earth has been accounted for while higher order effects and local geoid perturbations have been neglected. The purpose here is to derive equations for g_x, g_y and g_z that are consistent with this philosophy for modeling the earth. The normal component g_z will be tackled first following the approach beginning on page 78 of Reference 4.

■ Derivation for Normal Gravity, g_z

Define γ as gravity normal to the ellipsoid at altitude zero. Then for an altitude h above the ellipsoid, g_z at this altitude can be expanded in a MacLaurin series of terms in h:

$$g_\gamma = g_\gamma(\phi, h)$$

$$= g_\gamma(\phi, 0) + \left.\frac{\partial g_\gamma}{\partial h}\right|_{h=0} \cdot h + \frac{1}{2} \left.\frac{\partial^2 g_\gamma}{\partial h^2}\right|_{h=0} \cdot h^2 + \cdots$$

$$\triangleq \gamma + \frac{\partial \gamma}{\partial h} h + \frac{\partial^2 \gamma}{\partial h^2} h^2 + \cdots \qquad (56)$$

The first partial is given by Brun's formula (Reference 4, Equation 2-79) which is based on an ellipsoidal earth model:

$$\frac{\partial \gamma}{\partial h} = -\gamma\left(\frac{1}{R_m} + \frac{1}{R_p}\right) - 2\Omega^2 \qquad (57)$$

where R_m and R_p are the principle radii of curvature defined by (33) and (34). Taking reciprocals and expanding in a binomial series gives

$$\frac{1}{R_m} = \frac{(1-e^2\sin^2\phi)^{3/2}}{R_e(1-e^2)} = \frac{1}{R_e(1-e^2)}\left(1 - \frac{3}{2}e^2\sin^2\phi - \cdots\right)$$

$$\frac{1}{R_p} = \frac{(1-e^2\sin^2\phi)^{1/2}}{R_e} = \frac{1}{R_e}\left(1 - \frac{1}{2}e^2\sin^2\phi - \cdots\right)$$

Truncating these equations, adding them, and dropping higher order terms, produces the following result

$$\frac{1}{R_m} + \frac{1}{R_p} \cong \frac{1}{R_e}\left(2 + e^2 - 2e^2\sin^2\phi\right) \qquad (58)$$

If γ_e is the value of γ at the equator at h=o, the first order relationship between γ_e and Ω^2 is $\Omega^2 = m\gamma_e/R_e$ where m is 0.003449783. Substituting this and (58) in (57), and simplifying, yields

$$\frac{\partial \gamma}{\partial h} = -\frac{\gamma}{R_e}\left(2 + e^2 + m - 2e^2 \sin^2\phi\right) \tag{59}$$

The second ... tive $\partial^2\gamma/\partial h^2$ may be taken from the spherical approximation obtained when earth flattening is neglected entirely. Then according to Newton's law of mass attraction

$$\gamma = kM/R_e^2$$

where M is earth's mass and k is the universal gravitational constant.

$$\frac{\partial \gamma}{\partial h} = \frac{\partial \gamma}{\partial R_e} = -\frac{2kM}{R_e^3}$$

$$\frac{\partial^2 \gamma}{\partial h^2} = \frac{\partial^2 \gamma}{\partial R_e^2} = \frac{6kM}{R_e^4}$$

so that

$$\frac{\partial^2 \gamma}{\partial h^2} = \frac{6\gamma}{R_e^2} \tag{60}$$

Combining (59) and (60) with (56) produces the desired approximate equation for normal gravity.

$$g_z = \gamma\left[1 - \frac{1}{R_e}\left(2 + e^2 + m - 2e^2 \sin^2\phi\right)h + \frac{3}{R_e^2}h^2\right] \quad (61)$$

where γ is gravity at the ellipsoid surface which is given in Reference 1, page 22, as

$$\gamma = \gamma_e\left(1 + 0.005278994 \sin^2\phi + 0.000023461 \sin^4\phi\right) \quad (62)$$

$$\gamma_e = -32.0877057 \quad ft/sec^2 \quad (63)$$

Combining (62) and (63) with (61), and evaluating all constants, produced this final expression for g_z:

$$g_z = -\left[32.0877057 + 0.16939081 \sin^2\phi + 0.000752810 \sin^4\phi\right] \times$$

$$\times\left[1.0 - \left(9.6227E\text{-}8 - 6.4089E\text{-}10 \sin^2\phi\right)h + 6.8512E\text{-}15\ h^2\right] \quad (64)^*$$

■ Derivation for Level Gravity, g_x and g_y

At first it is somewhat surprising to realize that plumb-bob gravity has a level component. Such component arises because level surfaces at different altitudes (but same latitude) are not parallel. This fact is evident when one considers these two extremes: at

h=o the level surface is the ellipsoid and gravity points along ϕ; at the same latitude but elevated to $h=\infty$, gravity points at earths center of mass. Between these extremes the difference in slope of the two gravity vectors is the difference between geographic and geocentric latitude.

Another way to view the level gravity phenomenon is through the curvature of the normal plumb line as illustrated in Figure 14. Curvature is zero in the east-west direction owing to the rotational symmetry of the ellipsoid of revolution. Thus level gravity is entirely a north-south acceleration.

From Figure 14, observe the following relationship

$$dh = r\,d\beta \qquad (65)$$

The plumb line's radius of curvature, r, is given by (2-22a) in Reference 4:

$$r = \frac{1}{\dfrac{1}{g_\gamma} \cdot \dfrac{\partial g_\gamma}{\partial d}} \qquad (66)$$

where d is distance along a north-south direction. Combining (66) with (65) and rearranging

$$d\beta = \frac{1}{g_\gamma} \cdot \frac{\partial g_\gamma}{\partial d}\,dh \qquad (67)$$

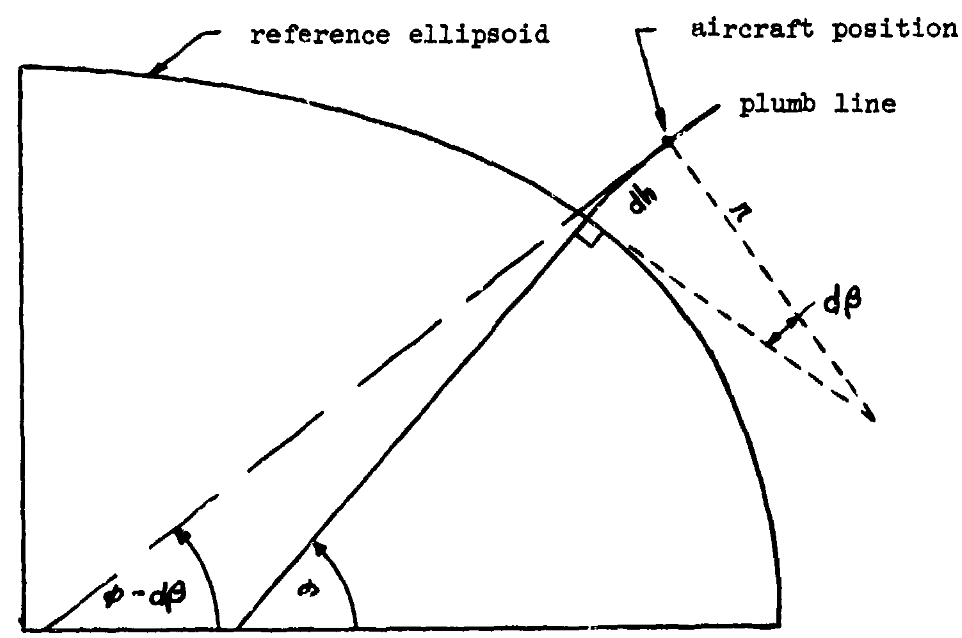


Figure 14 - Geometry for Deriving Level Gravity

The change in plumb line direction, β, between h=o and h=h is

$$\beta = \int_0^h \frac{1}{g_z} \cdot \frac{\partial g_z}{\partial d}\, dh \tag{68}$$

An approximate relationship for d is $d = R_e \phi$. Then

$$\frac{\partial g_z}{\partial d} = \frac{\partial g_z}{\partial \phi} \cdot \frac{\partial \phi}{\partial d} = \frac{\partial g_z}{\partial \phi} \cdot \frac{1}{R_e}$$

Thus (68) becomes'

$$\beta = \frac{1}{R_e} \int_0^h \frac{1}{g_z} \cdot \frac{\partial g_z}{\partial \phi}\, dh \tag{69}$$

To obtain a closed form expression for (69), simplify g_z as follows:

$$g_z = \gamma\left[1 - \frac{1}{R_e}(2 + e^2 + m - 2e^2 \sin^2\phi)h + \frac{3}{R_e^2} h^2\right] \quad (61)$$

$$\cong \gamma\left[1 - 2h/R_e\right]$$

$$\triangleq \gamma_e (1 + f_1 \sin^2\phi + f_2 \sin^4\phi)\left[1 - 2h/R_e\right] \quad \text{using (62)}$$

$$\cong \gamma_e (1 + f_1 \sin^2\phi - 2h/R_e)$$

Then

$$\frac{1}{g_3}\frac{\partial g_3}{\partial \phi} \cong \frac{1}{\gamma_e}\left(\gamma_e \, 2f_1 \sin\phi \cos\phi\right) = 2f_1 \sin\phi \cos\phi$$

Substitute this in (69) and integrate

$$\beta = \frac{1}{R_e}\int_0^h 2f_1 \sin\phi \cos\phi \, dh$$

$$= \frac{2f_1 \sin\phi \cos\phi}{R_e} h \qquad (70)$$

β is the tilt angle through which the gravity vector tips over as altitude increases. Projecting the magnitude of gravity (approximated here as $|\gamma_e|$) through β and onto the level surface gives for g_n (g north)

$$g_n = -|\gamma_e|\beta$$

$$= -1.63 \times 10^{-8}(h \sin\phi \cos\phi) \qquad (71)$$

Now rotate g_n through α to obtain g_x and g_y

$$g_x = g_n \cos\alpha = -1.63 \times 10^{-8} h \sin\phi \cos\phi \cos\alpha \qquad (72)$$

$$g_y = -g_n \sin\alpha = 1.63 \times 10^{-8} h \sin\phi \cos\phi \sin\alpha \qquad (73)$$

which may be stated in terms of the elements of C_e^n as

$$g_x = -1.63 \times 10^{-8}\ h\ CEN_{31}\ CEN_{11} \tag{74}^*$$

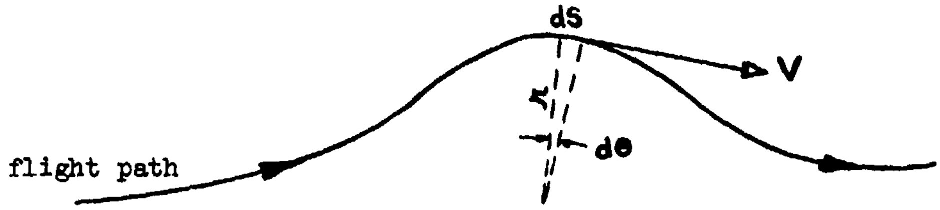
$$g_y = -1.63 \times 10^{-8}\ h\ CEN_{31}\ CEN_{21} \tag{75}^*$$

This derivation of level gravity was based on material in Section 5-6 of Reference 4 where it is pointed out that the effect of topographic irregularities on the curvature of the plumb line often overwhelms the value from equation (71). In high mountains the actual deflection could be 10 times greater so the limitations of (71) are apparent.

4.3 Path to Nav Rotation Rates and Control Equations

The relationships derived here for $\underline{\omega}$ will produce turning rates commensurate with the input data and with the restriction that level-plane turns be coordinated. In addition, equations for controlling the application of $\underline{\omega}$ will be derived. This control will usually take the form of a switch to turn $\underline{\omega}$ on or off at a critical event time. The control equation will compute the event time; e.g. the time at which $\dot{\eta}_y$ should be disabled in a vertical turn to make $\Delta\eta_y$ = PITCH This section evolved from the work in Section 3 of Reference 6.

As a preface, we list some basic kinematic equations for the illustration below where S is arc length, V is speed tangent to the path, r is radius of curvature and a_n is acceleration normal to the curved path:



$$V = \frac{dS}{dt} \qquad dS = r \cdot d\theta \qquad a_n = \frac{V^2}{r} \tag{76}$$

Combining these equations produces this relation for angular rate

$$\frac{d\theta}{dt} = \frac{a_n}{V} \tag{77}$$

4.3.1 A General Expression for $\underline{\omega}$

Now recall equations (13) and (28) defining C_p^n and $\underline{\omega}$:

$$C_p^n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} c\Psi_z & -s\Psi_z & 0 \\ s\Psi_z & c\Psi_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\Psi_y & 0 & s\Psi_y \\ 0 & 1 & 0 \\ -s\Psi_y & 0 & c\Psi_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\Psi_x & -s\Psi_x \\ 0 & s\Psi_x & c\Psi_x \end{bmatrix} \tag{13}$$

$$\triangleq T_{180} \cdot T_z \cdot T_y \cdot T_x \tag{78}$$

$$\underline{\omega} \triangleq \underline{\omega}_{pn}^{n} \triangleq \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{78}$$

where T_{180}, T_z, T_y and T_x are introduced here for reasons that will be apparent shortly.

Each Euler angle, η_x, η_y and η_z, has an associated rate, $\dot{\eta}_x$, $\dot{\eta}_y$ and $\dot{\eta}_z$. For a given path to nav orientation, the vector associated with $\dot{\eta}_x$ is directed along x_p. If the given path frame is rotated so that roll is zero (η_x=o), the vector associated with $\dot{\eta}_y$ is directed along the new (unrolled) y_p axis. When the new path frame is rotated again to remove pitch (η_y=o), the vector associated with $\dot{\eta}_z$ is directed along the new (unrolled and unpitched) z_p axis. Note that these three vectors <u>are not</u> mutually orthogonal. When transformed into the nav frame and added vectorially, they give the entire path to nav rotation velocity. Thus

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = T_{180} T_z T_y T_x \begin{pmatrix} \dot{\eta}_x \\ 0 \\ 0 \end{pmatrix} + T_{180} T_z T_y \begin{pmatrix} 0 \\ \dot{\eta}_y \\ 0 \end{pmatrix} + T_{180} T_z \begin{pmatrix} 0 \\ 0 \\ \dot{\eta}_z \end{pmatrix} \tag{79}$$

This may be simplified using $\eta_z = \alpha + \psi$ (Figure 11) and the definitions of T_{180}, T_z and C_p^n in (78):

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = C_p^n \begin{pmatrix} \dot{\eta}_x \\ 0 \\ 0 \end{pmatrix} + T_{180} T_z T_y \begin{pmatrix} 0 \\ \dot{\eta}_y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\dot{\alpha} - \dot{\psi} \end{pmatrix} \qquad (80)$$

Equation (80) is the most general relation for angular rate between the path and nav frames. It will simplify considerably depending on (1) the type of maneuver (2) the nominal path (great circle or rhumb line) over which that maneuver is superimposed, and (3) $\dot{\alpha}$ which is given in Table 2 as a function of the nav frame mechanization choice. In the following four subsections it is assumed that the reader is familiar with Section 3.4.

4.3.2 Vertical Turn

a. $\underline{\omega}$ Equation

Since the aircraft's wings remain level in a vertical turn, $T_x = I$ and $\dot{\eta}_x = 0$. Thus (80) becomes

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = T_{180} T_z T_y I \begin{pmatrix} 0 \\ \dot{\eta}_y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\dot{\alpha} - \dot{\psi} \end{pmatrix}$$

or

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = C_p^n \begin{pmatrix} 0 \\ \dot{\eta}_y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\dot{\alpha} - \dot{\psi} \end{pmatrix} \qquad (81)^*$$

where $\dot{\eta}_y$ is given by (77) as

$$\dot{\eta}_y = \frac{a_n}{V_T(t)} \qquad (82)^*$$

and

$$V_T(t) = V_T(t_i) + (t - t_i)\,\dot{V}_T \qquad (83)$$

$$a_n = TACC \cdot \mathrm{sign}\,(PITCH) \qquad (84)^*$$

Note that TACC is positive and in units of ft/sec^2. A vertical turn will have a slight heading rate if the aircraft is following a great circle path so

$$\dot{\psi} = \dot{\psi}_N \triangleq \begin{cases} 0 & \text{, rhumb line} \\ \dot{\psi}_G & \text{, great circle (Section 4.3.6)} \end{cases} \qquad (85)^*$$

b. Control Derivation

Equation (82) can be integrated to yield change in η_y over the interval (t_i, t). In the case where V_T varies linearly with time ($\dot{V}_T$ = PACC $\neq 0$),

$$\Delta \eta_y(t) = \int_{t_i}^{t} \dot{\eta}_y(\tau)\,d\tau = \int_{t_i}^{t} \frac{a_n}{V_T(\tau)}\,d\tau$$

$$= \int_{t_i}^{t} \frac{a_n}{V_T(t_i) + (\tau - t_i)\dot{V}_T}\,d\tau$$

$$= \frac{a_n}{\dot{V}_T} \ln\left[1 + \frac{\dot{V}_T}{V_T(t_i)}(t - t_i)\right], \quad \dot{V}_T \neq 0 \qquad (86)$$

In the case where V_T is constant

$$\Delta \eta_y(t) = \int_{t_i}^{t} \frac{a_n}{V_T}\,d\tau = \frac{a_n}{V_T}(t - t_i), \quad \dot{V}_T = 0 \qquad (87)$$

Equations (86) and (87) may be inverted to compute a time, t = TDONE, when $\Delta\eta_y$ (TDONE) = |PITCH|:

$$TDONE = \begin{cases} t_i + \dfrac{V_T(t_i)}{\dot{V}_T}\left[\exp\left(\dfrac{\dot{V}_T \cdot PITCH}{a_n}\right) - 1\right], & \dot{V}_T \neq 0 \\ t_i + \dfrac{PITCH}{a_n} V_T, & \dot{V}_T = 0 \end{cases} \qquad (88)^*$$

4.3.3 Horizontal Turn

a. $\underline{\omega}$ Equation

Since the aircraft does not pitch in a horizontal turn, $\dot{\eta}_y$ is zero and (80) becomes

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = C_p^n \begin{pmatrix} \dot{\eta}_x \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\dot{\alpha} - \dot{\psi} \end{pmatrix} \qquad (89)^*$$

where η_x behaves as pictured in Figure 5. ($\dot{\eta}_x$ is either on or off. When on, $\dot{\eta}_x$ = ± ROLRATE.) $\dot{\psi}$ is the sum of $\dot{\psi}_N$, the nominal path contribution from (85), and $\dot{\psi}_M$, the maneuver contribution due to TACC:

$$\dot{\psi} \triangleq \dot{\psi}_M + \dot{\psi}_N \qquad (90)$$

$$= \begin{cases} \dot{\psi}_M & \text{, rhumb line} \\ \dot{\psi}_M + \dot{\psi}_G & \text{, great circle} \end{cases} \qquad (91)^*$$

● Coordinated Turn Requirement

During the turn the normal acceleration, $a_n(t)$, progresses from zero to a peak - a flat peak has a magnitude of TACC ft/sec^2 - and back to zero (see Figure 6). This progression occurs because $a_n(t)$

must "follow" $\eta_x(t)$ to satisfy the requirement for coordinated turns. This requirement manifests itself in this way:

$$a_n(t) = 32.2 \cos \eta_y \tan [\eta_x(t)] \qquad (92)$$

Equation (92) shows the aircraft will turn only if its wings are banked. The genesis for (92) is provided in Figure 15, a nose-on view of the aircraft in a right turn with pitch zero ($\eta_y = 0$).

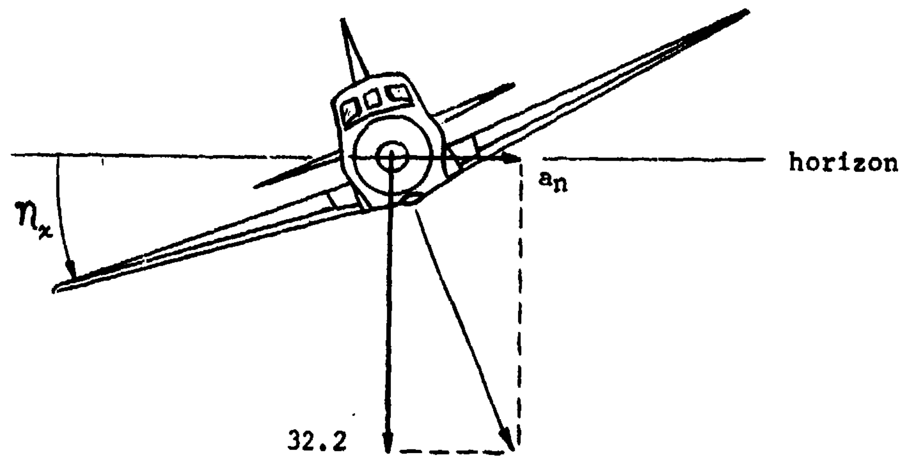


Figure 15 - Balancing Accelerations in a Coordinated Turn

The vector sum of 32.2 and a_n must act perpendicular to the wings in order to implement the coordinated turn. Thus

$$a_n = 32.2 \tan \eta_x \qquad (93)$$

When pitch is nonzero, Figure 15 is altered by making the downward component of gravity $32.2 \cdot \cos \eta_y$ instead of 32.2. Equation (92) then follows immediately.

• $\dot{\psi}_M$ Equation

Defining V_L as the level-plane component of total speed ($V_L = V_T \cos \eta_y$), we may plug (92) in (77) to get the maneuver turning rate:

$$\dot{\psi}_M = \frac{32.2 \cos \eta_y \tan[\eta_x(t)]}{V_L(t)}$$

$$= \frac{32.2 \tan[\eta_x(t)]}{V_T(t)} \qquad (94)^*$$

$$= \frac{32.2 \tan[\eta_x(t)]}{V_T(t_i) + (t-t_i)\dot{V}_T} \qquad (95)$$

b. Control Derivation

Examination of (89) and (94) shows that roll and roll rate must be known before $\underline{\omega}$ can be computed. Their determination rests on choosing the appropriate roll history from Figure 5 and then on computing TOFF, TON and TDONE. The logic and calculations for accomplishing this are contained in PROFGEN subroutines TSETUP2 and YAWCHG and are outlined below.

● $\Delta\psi_M$ Computation

The most general roll history is pictured in Figure 16. (This roll history is for a right turn. A left turn would be the negative of Figure 16).

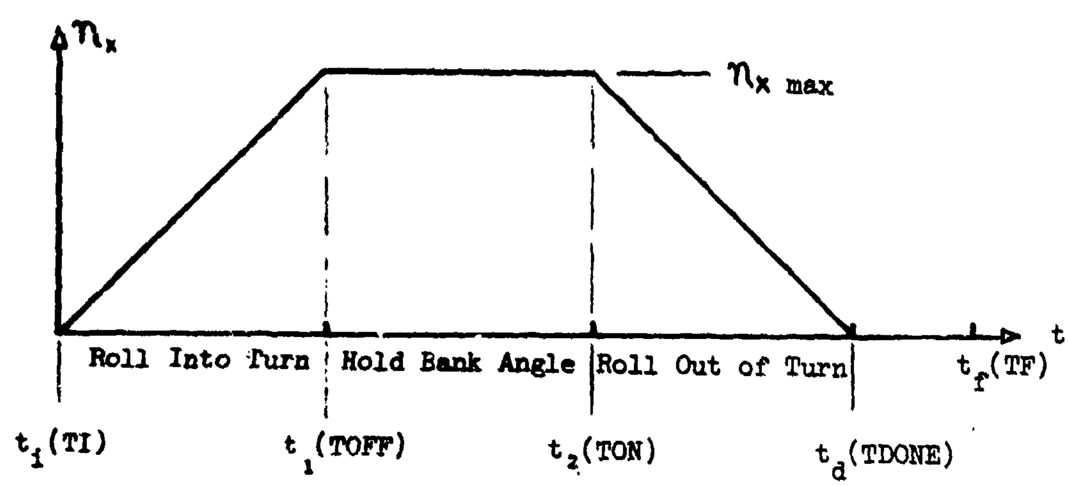


Figure 16 - Roll Angle History (Case A)

We wish to compute the change in heading, $\Delta\psi_M$, that would occur if Figure 16 was the roll history. For this purpose we may assume t_1, t_2 and t_d are time increments measured from a t_i of zero. Set up the integral of (94) as follows:

$$\Delta\psi_M = 32.2\left\{\int_0^{t_1}\frac{\tan[n_x(\tau)]}{V_T(\tau)}d\tau + \int_{t_1}^{t_2}\frac{\tan[n_{x_{max}}]}{V_T(\tau)}d\tau + \int_{t_2}^{t_d}\frac{\tan[n_x(\tau)]}{V_T(\tau)}d\tau\right\} \qquad (96)$$

From (92)

$$\tan \eta_{x_{max}} = \frac{|a_{n_{max}}|}{32.2 \cos \eta_y} = \frac{TACC}{32.2 \cos \eta_y} \qquad (97)$$

Recalling (83) for $V_T(t)$, the middle integral in (96) is

$$\int_{t_1}^{t_2} \frac{\tan \eta_{x_{max}}}{V_T(\tau)} d\tau = \begin{cases} \dfrac{TACC}{32.2 \cos \eta_y \, \dot{V}_T} \ln\left[1 + \dfrac{(t_2 - t_1)\dot{V}_T}{V_T(t_1)}\right], & \dot{V}_T \neq 0 \\ \dfrac{TACC}{32.2 \cos \eta_y \, V_T} (t_2 - t_1), & \dot{V}_T = 0 \end{cases} \qquad (98)$$

The first and third integrals in (96) are not closed-form integrable unless $\dot{V}_T$=0. Satisfactory approximations have been obtained for them by substituting $\overline{V}_T$, average speed, for $V_T(t)$ as follows:

$$\int_0^{t_1} \frac{\tan \eta_x(\tau)}{V_T(\tau)} \cong \frac{1}{\overline{V}_{T1}} \int_0^{t_1} \tan \eta_x(\tau) d\tau$$

$$= \frac{1}{\overline{V}_{T1}} \int_0^{t} \tan(\dot{\eta}_x \tau) d\tau$$

$$= \frac{-\ln\left[\cos(\dot{\eta}_x t_1)\right]}{\overline{V}_{T1} \dot{\eta}_x} \qquad (99)$$

where $\dot{\eta}_x$ = ROLRATE and V_{T1}, average speed in (t_i, t_1), is

$$\bar{V}_{T1} = \frac{V_T(t_i) + V_T(t_1)}{2} = V_T(t_i) + \frac{t_1 \dot{V}_T}{2} \qquad (100)$$

Similarly

$$\int_{t_2}^{t_d} \frac{\tan \eta_x(\tau)}{V_T(\tau)} d\tau \cong \frac{-\ln[\cos(\dot{\eta}_x t_1)]}{\bar{V}_{Td}\, \dot{\eta}_x} \qquad (101)$$

$$\bar{V}_{Td} = \frac{V_T(t_2) + V_T(t_d)}{2} = V_T(t_i) + \left(t_2 + \frac{t_1}{2}\right)\dot{V}_T \qquad (102)$$

Inserting (98) - (101) in (96) and simplifying gives

$$\Delta\psi_M \cong \begin{cases} \dfrac{-32.2}{\dot{\eta}_x}\left\{ \dfrac{\ln[\cos(\dot{\eta}_x t_1)]}{V_T(t_i) + \dfrac{t_1 \dot{V}_T}{2}} + \dfrac{\ln[\cos(\dot{\eta}_x t_1)]}{V_T(t_i) + \left(t_2 + \dfrac{t_1}{2}\right)\dot{V}_T} \right\} + \\ \quad + \dfrac{TACC \ln\left[1 + \dfrac{(t_2 - t_1)\dot{V}_T}{V_T(t_1)}\right]}{\cos \eta_y \, \dot{V}_T} \qquad , \quad \dot{V}_T \neq 0 \\ -\dfrac{64.4 \ln[\cos(\dot{\eta}_x t_1)]}{\dot{\eta}_x V_T} + \dfrac{TACC\,(t_2 - t_1)}{\cos\eta_y V_T} \qquad , \quad \dot{V}_T = 0 \end{cases} \qquad (103)^*$$

Since η_{xmax} and $\dot{\eta}_x$ (=ROLRATE) are known, t_1 is

$$t_1 = \frac{\eta_{x\,max}}{\dot{\eta}_x} = \frac{\tan^{-1}(TACC/32.2\cos\eta_y)}{\dot{\eta}_x} \qquad (104)^*$$

- Reasoning on $\Delta\psi_{Mmax}$

PROFGEN determines if the maneuver can be completed (HEAD reached) by seeing how far the aircraft would turn if turn acceleration was left on for the entire segment, t_i to t_f. Equation (103) is used for this purpose where t_1 is obtained from (104) and t_2 is placed t_1 seconds short of $t_d = t_f$. (If 2 t_1 exceeds SEGLNT, t_1 is set to SEGLNT ÷ 2.) PROFGEN solves for $\Delta\psi_{Mmax}$ in subroutine YAWCHG using (103).

If $\Delta\psi_{Mmax}$ exceeds |HEAD|, the turn can be completed and either Case A or B of Figure 5 is appropriate. Having decided A or B (not C or D), the problem becomes determination of t_1 and t_2 (In Case B, $t_1 = t_2$.) The following paragraphs will derive equations for t_1 and t_2 for both Case A and Case B.

If $\Delta\psi_{Mmax}$ falls short of |HEAD|, the turn cannot be completed and either Case C or D of Figure 5 is appropriate. For Cases C and D, the determination of t_1 and t_2 is trivial.

- Case A Roll History

The roll history shown in Figure 16 is identifiable as Case A from Figure 5. Setting $\Delta\psi_M$ = |HEAD| and obtaining t_1 from (104), everything is known in (103) except t_2, the time at which roll-out

should begin. Unfortunately, (103) cannot be easily inverted for t_2. This difficulty was overcome by ridding (103) of its "ln" function which was accomplished by approximating the middle integral of (96) just as the first and last integrals in (96) were approximated earlier. Thus (98) becomes

$$\int_{t_1}^{t_2} \frac{\tan \eta_{x\,max}}{V_T(\tau)}\, d\tau = \frac{TACC\,(t_2 - t_1)}{32.2 \cos \eta_y \, \bar{V}_{T2}} \qquad (105)$$

where

$$\bar{V}_{T2} = V(t_i) + \frac{t_1 + t_2}{2} \dot{V}_T \qquad (106)$$

Now replace the first term in (103) with (105) - (106), set $\Delta\psi_M = |HEAD|$ and simplify to get this quadratic equation in t_2:

$$t_2^2 \left[\left(\tfrac{1}{2}\right) b_1 \dot{V}_T + \left(\tfrac{1}{2}\right) b_0 \dot{V}_T / b_2 - TACC \right] \dot{V}_T$$

$$+ t_2 \left[\left(\tfrac{3}{2}\right) b_1 b_2 \dot{V}_T + 2 b_0 \dot{V}_T - TACC\, b_2 + TACC\, t_1 \dot{V}_T \right]$$

$$+ b_2 \left[b_1 b_2 + 2 b_0 + TACC\, t_1 \right] = 0 \qquad (107)^*$$

where

$$\left.\begin{aligned} b_0 &= 32.2 \cos \eta_y \ln\left[\cos(\dot{\eta}_x t_1)\right] \div \dot{\eta}_x \\ b_1 &= |HEAD| \cos \eta_y \\ b_2 &= V(t_i) + t_1 \dot{V}_T/2 \end{aligned}\right\} \quad (108)^*$$

Note that (107) reduces to a linear equation in t_2 if $\dot{V}_T$ is zero.

The coefficients in (107) are computed in TSETUP2 and supplied to QUADRT where t_2 is computed. TOFF, TON and TDONE are given below. To reference them to true time instead of $t_i=0$, merely add TI to each one.

$$TOFF = t_1 = \tan^{-1}\left[TACC/(32.2 \cos \eta_y)\right]/\dot{\eta}_x \qquad (109)^*$$

$$TON = t_2 = \text{solution of } (107)$$

$$TDONE = t_d = t_2 + t_1$$

- Case B Roll History

A "Case B" type roll history is illustrated below.

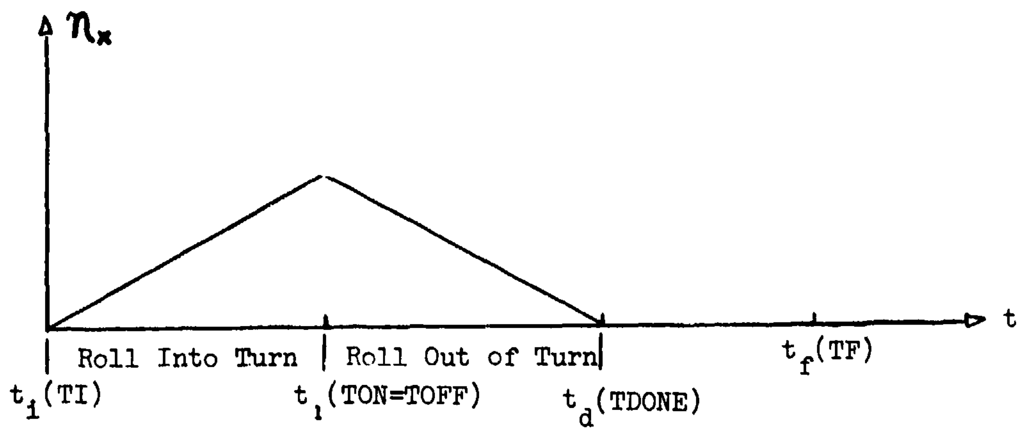


Figure 17 - Roll Angle History (Case B)

The following equation in t_1 was obtained using a procedure like that which lead to (103):

$$64.4 \ln\left[\cos(\dot{\eta}_x t_1)\right] + \dot{\eta}_x \left|HEAD\right| \left[\dot{V}_T t_1 + V_T(t_i)\right] = 0 \qquad (109)$$

If ln (cos x) is approximated as $-.632x^2$, (109) becomes

$$t_1^2\left[-40.7\,\dot{\eta}_x^2\right] + t_1\left[\dot{V}_T \left|HEAD\right| \dot{\eta}_x\right] + V_T(t_i)\left|HEAD\right|\dot{\eta}_x = 0 \qquad (110)^*$$

The coefficients in (110) are computed in TSETUP2 and supplied to QUADRT where t_1 is computed. TOFF, TON and TDONE follow immediately (see Figure 17) when t_1 is known.

4.3.4 Sine Maneuver

a. $\underline{\omega}$ Equation

Since the aircraft does not change pitch in a sine maneuver, η_y is zero. Hence $\underline{\omega}$ in (80) reduces to a form identical to that for a horizontal turn:

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = C_p^n \begin{pmatrix} \dot{\eta}_x \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\dot{\alpha} - \dot{\psi} \end{pmatrix} \qquad (89)^*$$

$\dot{\psi}$ is the sum of $\dot{\psi}_N$, the nominal path contribution from (85), and $\dot{\psi}_M$, the maneuver contribution due to $a_n(t)$:

$$\dot{\psi} \triangleq \dot{\psi}_M + \dot{\psi}_N \tag{90}$$

$$= \begin{cases} \dot{\psi}_M & \text{, rhumb line} \\ \dot{\psi}_M + \dot{\psi}_G & \text{, great circle} \end{cases} \tag{91}^{*}$$

The next two paragraphs derive expressions for $\dot{\psi}_M$ and $\dot{\eta}_x$. Expressions for $\dot{\alpha}$ and $\dot{\psi}$ are given in Table 2 and Section 4.3.6, respectively.

b. $\dot{\psi}_M$ Equation

For the aircraft to fly a sine maneuver, its heading must vary per Equation (5):

$$\dot{\psi}_M(t) = \begin{cases} +A \sin^2 wt & , t_i \le t < T_p/2 \\ -A \sin^2 wt & , T_p/2 \le t < T_p \end{cases} \tag{5}$$

where

A = max heading variation

w = heading oscillation frequency

T_p = $2\pi/w$ = period of one full oscillation

Differentiating (5) twice gives

$$\dot{\psi}_M = \begin{cases} A\omega \sin(2\omega t) \\ -A\omega \sin(2\omega t) \end{cases} \qquad (111)^*$$

$$\ddot{\psi}_M = \begin{cases} 2A\omega^2 \cos(2\omega t) \\ -2A\omega^2 \cos(2\omega t) \end{cases} \qquad (112)^*$$

c. $\dot{\eta}_x$ Equation

In the context of a sine maneuver, (77) becomes

$$\dot{\psi}_M = \frac{a_n(t)}{V_L(t)} = \frac{a_n(t)}{\cos\eta_y \; V_T(t)} \qquad (113)^*$$

where $a_n(t)$ acts in the level plane and V_L is level-plane speed. Now recall (92), the coordinated turn requirement relating $a_n(t)$ to $\eta_x(t)$.

$$a_n(t) = 32.2 \cos\eta_y \tan\left[\eta_x(t)\right] \qquad (92)$$

Plug (92) into (113) and rearrange to get

$$\eta_x(t) = \tan^{-1}\left(\frac{V_T(t)}{32.2}\,\dot{\psi}_M\right) \qquad (114)$$

from which it follows that

$$\dot{n}_x = \frac{32.2\ V_T}{32.2^2 + (V_T\ \dot{\psi}_M)^2}\ \ddot{\psi}_M \qquad (115)^*$$

The trouble that was experienced in establishing roll control in the horizontal turn is entirely avoided in the sine maneuver because there is no need to compute any special "event times".

4.3.5 Straight Flight

a. $\underline{\omega}$ Equation

Since aircraft attitude remains fixed in straight flight, (80) simplifies to

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\dot{\alpha} - \dot{\psi} \end{pmatrix} \qquad (116)^*$$

where

$$\dot{\psi} = \dot{\psi}_N = \begin{cases} 0 & \text{, rhumb line} \\ \dot{\psi}_G & \text{, great circle} \end{cases} \qquad (85)^*$$

To repeat, $\dot{\alpha}$ is given in Table 2 . As with the sine maneuver, no "events" occur in straight flight so (116) tells the whole story.

4.3.6 Heading Angle Turning Rate for a Great Circle Path

Figure 18 shows the geometry associated with the problem of determining the rate of change of heading along a great circle route. (The E frame in Figure 18 is established here to facilitate this analysis). A great circle route lies in a single plane, Plane I, which passes through the center of the earth. This plane is described by λ_{eq} and ψ_{eq} where λ_{eq} is the longitude at which the great circle plane, Plane I, intersects the equatorial plane and ψ_{eq} is the heading at the aforementioned intersection.

Consider a vehicle at point P proceeding along a path lying in Plane I. The coordinates of this point are given by $\lambda-\lambda_{eq}$, ϕ_c and R where ϕ_c is the geocentric latitude and R is the length of the geocentric radius vector.

The geocentric heading, ψ_c, at point P is given as the angle between the horizontal velocity vector and a vertical plane, Plane II, erected at longitude $\lambda-\lambda_{eq}$ and containing point P. Thus ψ_c is the angle between Planes I & II. In rectangular coordinates (X_E, Y_E, Z_E) the equations for Planes I & II are respectively

$$X_E - Y_E \tan\psi_{eq} = 0 \qquad (117)$$

$$X_E - Z_E \tan(\lambda - \lambda_{eq}) = 0 \qquad (118)$$

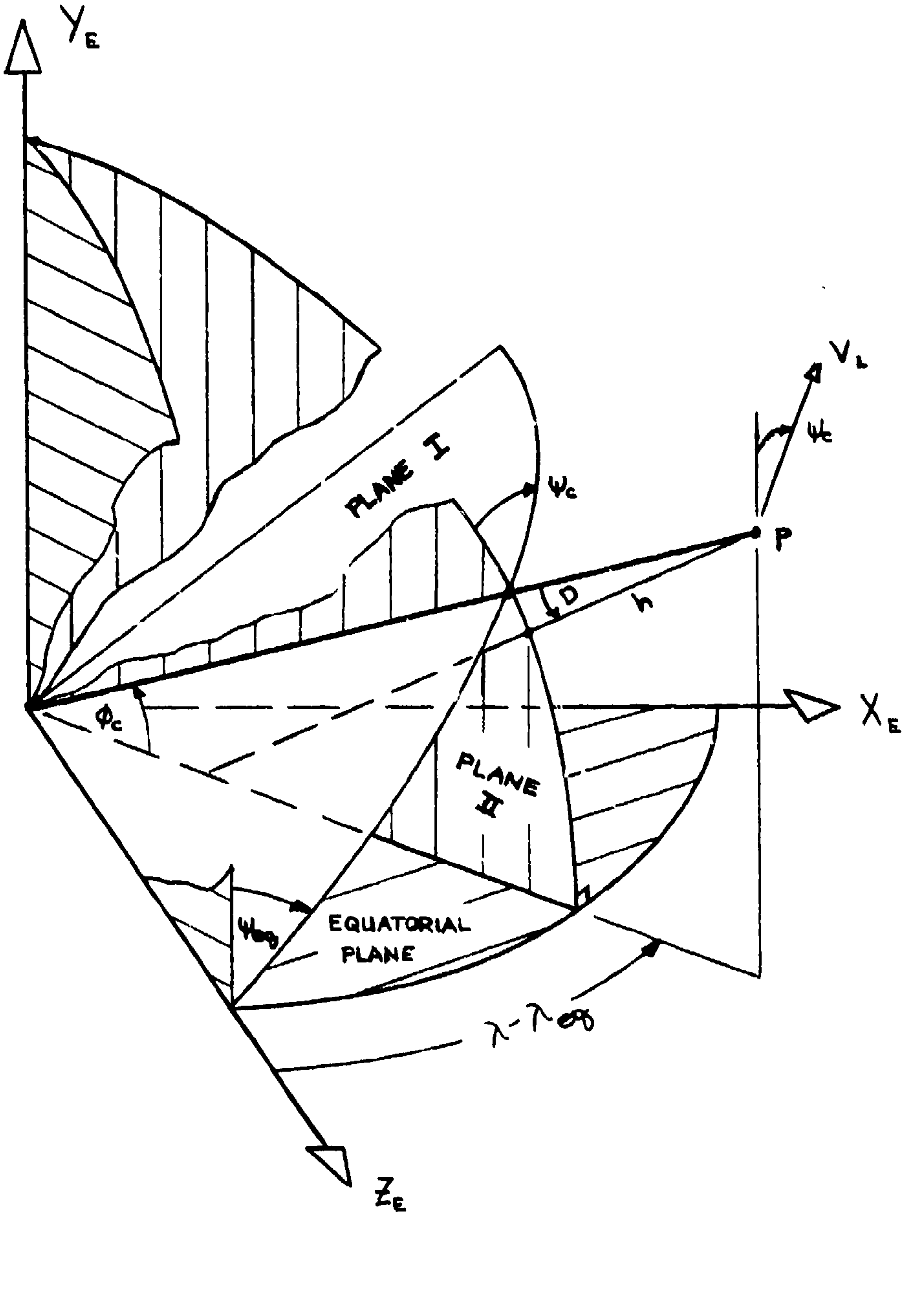


Figure 18 - Great Circle Geometry

The angle is therefore given by

$$\cos \psi_c = \cos \psi_{eq} \cos (\lambda - \lambda_{eq}) \qquad (119)$$

The primary concern here is with the geographic heading angle ψ rather than the geocentric heading ψ_c. These two angles differ due to the deviation angle D between the local vertical and the geocentric position vector, see Figure 18. If ϕ denotes geographic latitude, then

$$D = \phi - \phi_c \qquad (120)$$

The projection of ψ_c onto a local level coordinate system (rotation through angle D about the local east axis) yields

$$\sin \psi = \frac{\sin \psi_c}{\sqrt{1 - \cos^2 \psi_c \sin^2 D}} \qquad (121)$$

and

$$\cos \psi = \frac{\cos \psi_c \cos D}{\sqrt{1 - \cos^2 \psi_c \sin^2 D}} \qquad (122)$$

Differentiation of (121) with respect to time yields the desired quantity:

$$\dot{\psi} = \frac{\dot{\psi}_c \cos D + \dot{D} \sin \psi_c \cos \psi_c \sin D}{1 - \cos^2 \psi_c \sin^2 D} \qquad (123)$$

The remaining steps are concerned with determining usable expressions for the right side of (123).

The time derivative of ψ_c is obtained from equation (119) as

$$\dot{\psi}_c = \frac{\dot{\lambda}\sqrt{\cos^2\psi_{eq} - \cos^2\psi_c}}{\sin\psi_c} \tag{124}$$

From Figure 18 it is seen that

$$Y_E = R\sin\phi_c \tag{125}$$

and

$$Z_E = R\cos\phi_c \cos(\lambda - \lambda_{eq}) \tag{126}$$

Combining equations (117), (118), (119), (125) and (126) yields

$$\cos^2\psi_{eq} = \cos^2\psi_c \cos^2\phi_c + \sin^2\phi_c \tag{127}$$

which when substituted into (124) gives the simple expression

$$\dot{\psi}_c = \dot{\lambda}\sin(\phi - D) \tag{128}$$

Also required is the inverse solution of equations (121) and (122)

$$\sin\psi_c = \frac{\sin\psi\cos D}{\sqrt{\cos^2\psi\sin^2 D + \cos^2 D}} \tag{129}$$

$$\cos\psi_c = \frac{\cos\psi}{\sqrt{\cos^2\psi\sin^2 D + \cos^2 D}} \tag{130}$$

Substituting (128), (129) and (130) into (123) yields the expression for the turning rate of the heading angle

$$\dot{\psi} = \dot{\lambda} \sin(\phi - D) \cos D \left[1 + \cos^2\psi \tan^2 D\right] + \dot{D} \sin\psi \cos\psi \tan D \qquad (131)^*$$

Since this is the value of $\dot{\psi}$ required to maintain flight in the great circle plane, it is the quantity labeled previously as $\dot{\psi}_G$. Thus

$\dot{\psi}_G \Longleftrightarrow$ Equation (131)

The angle D and its time derivative $\dot{D}$ are given for the ellipsoidal earth by the following relationships:

$$\tan D = \frac{R_e\, e^2 \sin\phi \cos\phi}{R\sqrt{1 - \left[1 + (R_e\, e \cos\phi / R)^2\right] e^2 \sin^2\phi}} \qquad (132)^*$$

$$R = R_e \sqrt{\frac{1 - (2 - e^2)\, e^2 \sin^2\phi}{1 - e^2 \sin^2\phi} + 2\sqrt{1 - e^2 \sin^2\phi}\left(\frac{h}{R_e}\right) + \left(\frac{h}{R_e}\right)^2} \qquad (133)^*$$

$$\dot{D} = \frac{R_e\, e^2}{R \cos D \sqrt{1 - e^2 \sin^2\phi}} \left[\frac{\cos^2\phi - (1 - e^2 \sin^2\phi)\sin^2\phi}{1 - e^2 \sin^2\phi}\,\dot{\phi} - \frac{\dot{R}}{R} \sin\phi \cos\phi\right] \qquad (134)^*$$

$$\dot{R} = \frac{R_e}{R}\left[\dot{h}\left(\sqrt{1 - e^2 \sin^2\phi} + \frac{h}{R_e}\right) - \dot{\phi} R_e\, e^2 \sin\phi \cos\phi \left(\frac{1 - e^2}{(1 - e^2 \sin^2\phi)^2} + \frac{h/R_e}{\sqrt{1 - e^2 \sin^2\phi}}\right)\right]$$

$$(135)^*$$

Equations (131) through (135) are the exact relationships for an ellipsoidal earth. If the earth had been assumed spherical, its eccentricity would have been zero and (131) through (135) would reduce to,

$$\begin{aligned} \dot{\psi}_G &= \dot{\lambda} \sin \phi \\ D &= 0 \\ R &= R_e + h \\ \dot{D} &= 0 \\ \dot{R} &= \dot{h} \end{aligned} \tag{136}$$

The earth model in PROFGEN may be converted from an ellipsoid to a sphere by simply setting $e^2=0$ in BLOCK DATA. However, to take full advantage of the spherical-earth simplification would require replacing (131) through (135) with (136) and revising the gravity and earth radii computations.

SECTION V

PROGRAM ORGANIZATION

Previous sections have described what PROFGEN does, explained how to use it, and derived equations for its implementation. This section assembles these equations in a sequence amenable to solution in FORTRAN code. Flow of equations and code are both presented.

Two principles will guide us now (Reference 7):

- The most reliable documentation for any program is the code itself. Therefore our purpose is not to describe the code in minute detail - such a description would be unreliable, redundant, and probably harder to read than the code itself - but merely to show how large pieces of code interact.

- Each subprogram contains comments giving a readable description of what that subprogram is supposed to do. These comments form the core of the micro-level documentation and do not need to be repeated here.

Figure 19 is a macro-level flow chart emphasizing overall computational structure, especially with regard to control of step size, h. The name(s) beside each block in Figure 19 designates the subprogram(s) where the action in that block occurs. Each of these subprograms usually calls one or more other subprograms to complete

this action (see Figure 21). The main program and master executive is named PROFGEN. The subexecutive for controlling numerical integration during each maneuver is FLTPATH.

Figure 20 is an expansion of the integration block that appears in heavy outline in Figure 19. Figure 20 was included here to show how the differential equations in Section IV actually get solved.

Figure 21 is a dependency chart showing what calls what. Although timing relationships are vague, (Figures 19 and 20 deal with timing) this chart is nevertheless useful for getting a bigger picture of how PROFGEN fits together. It was kept during program development to help assess the impact of proposed changes.

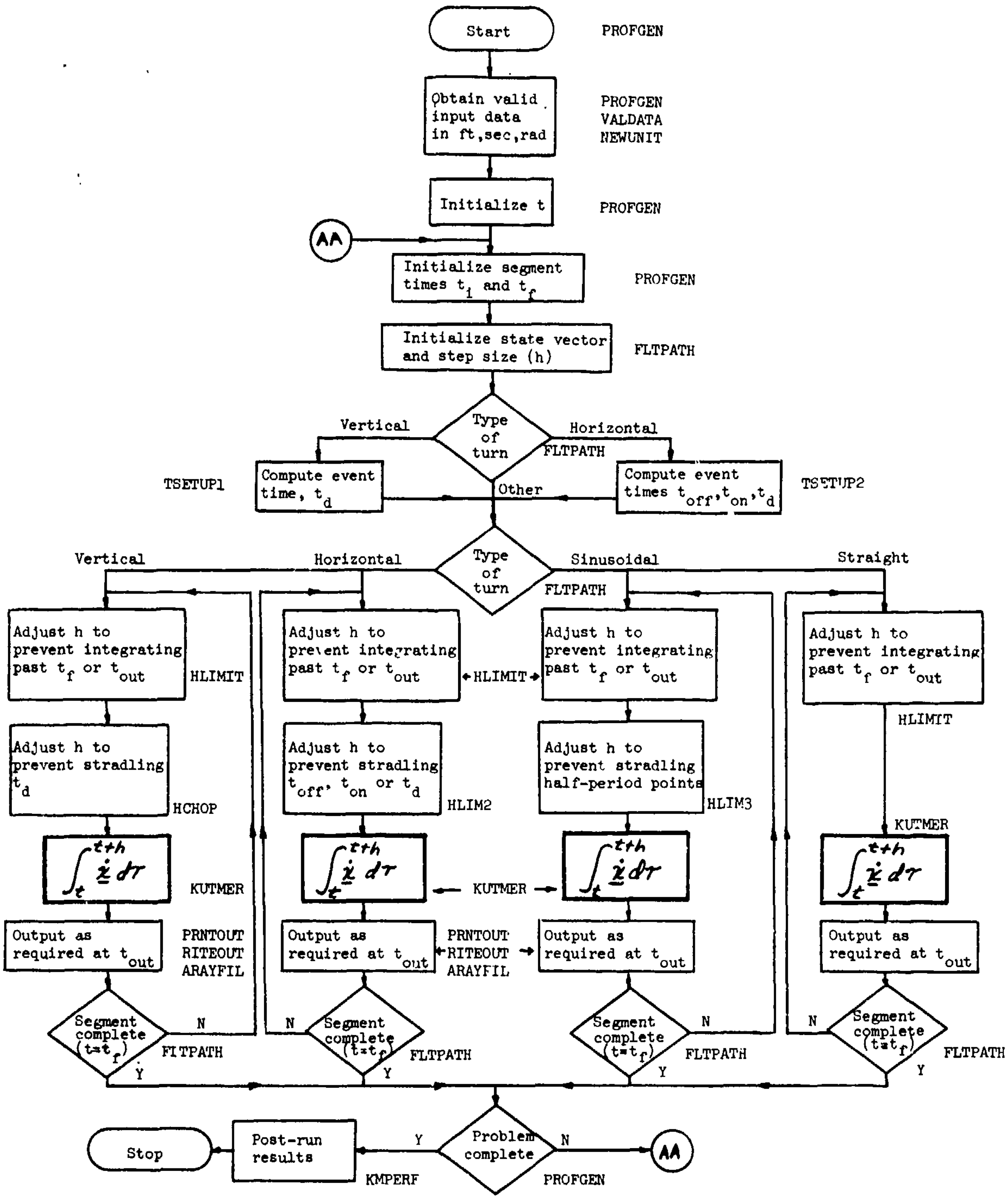


Figure 19 - Macro-Level Logic Flow Diagram

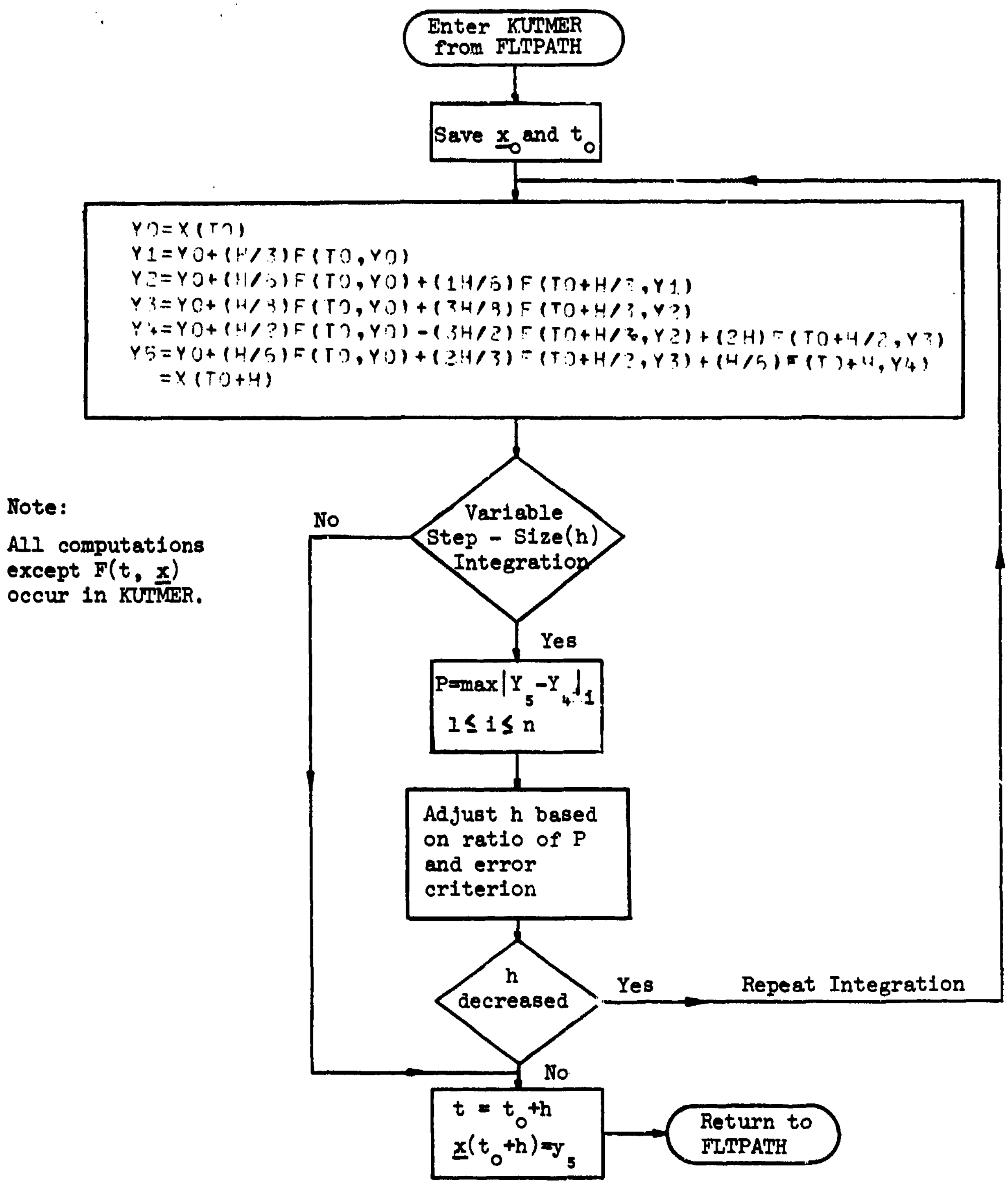


Figure 20 - Numerical Integration of $\dot{\underline{x}} = F(t, \underline{x})$ from t_o to t_o+h

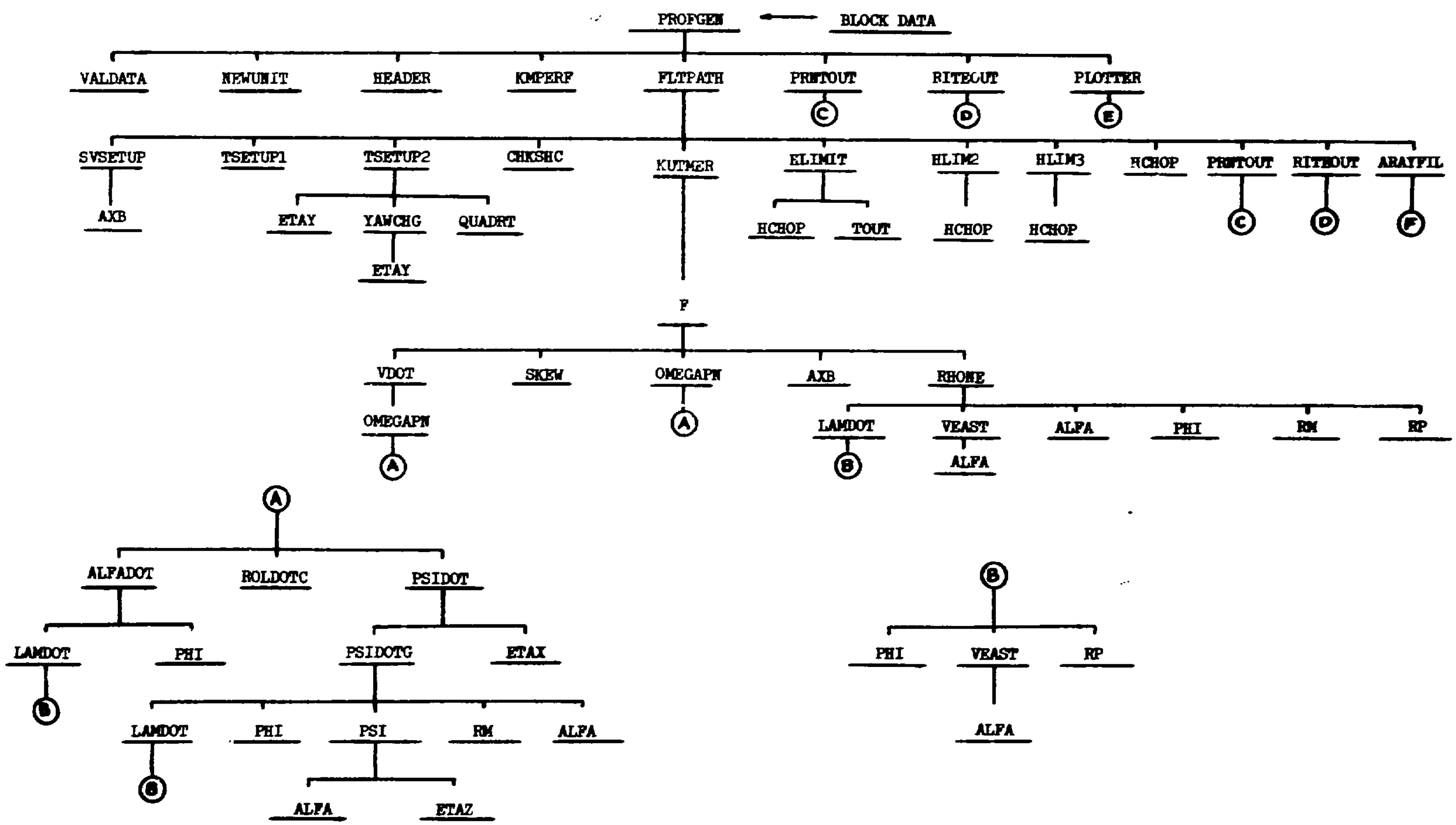
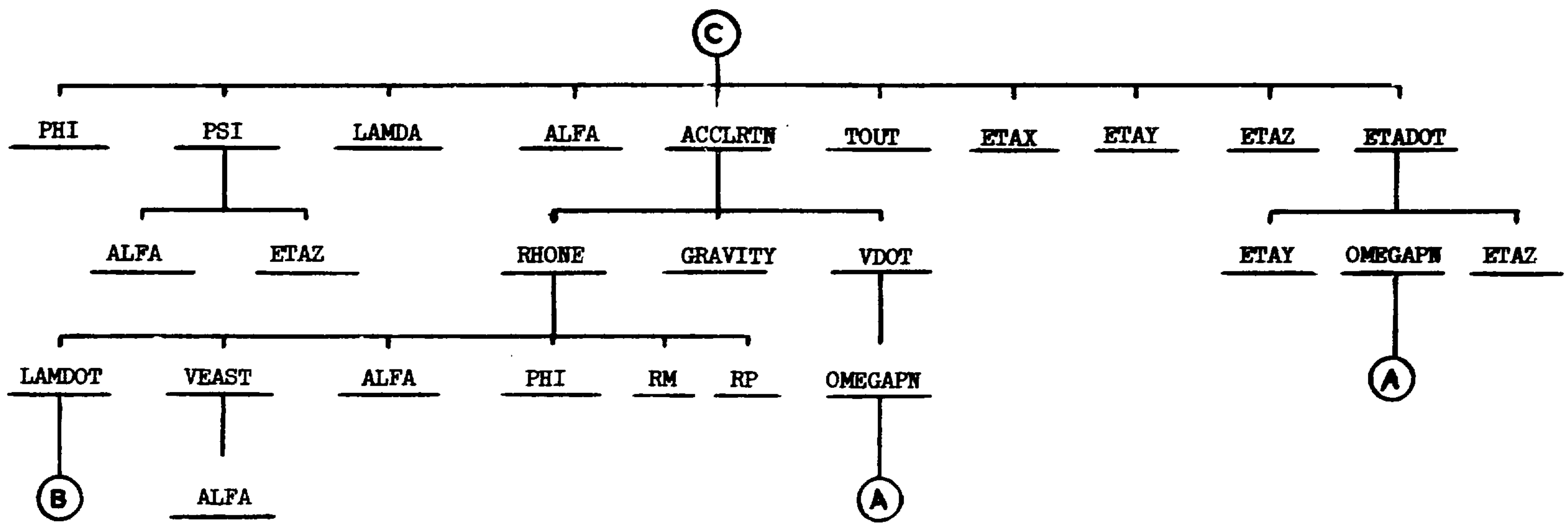


Figure 21 - Subprogram Dependency Chart



D ~ Same as C with calls to PSI and ETADOT removed.

F ~ Same as C with calls to PSI, ALFA, ACCLRTN and ETADOT removed.

Figure 21 (Continued)

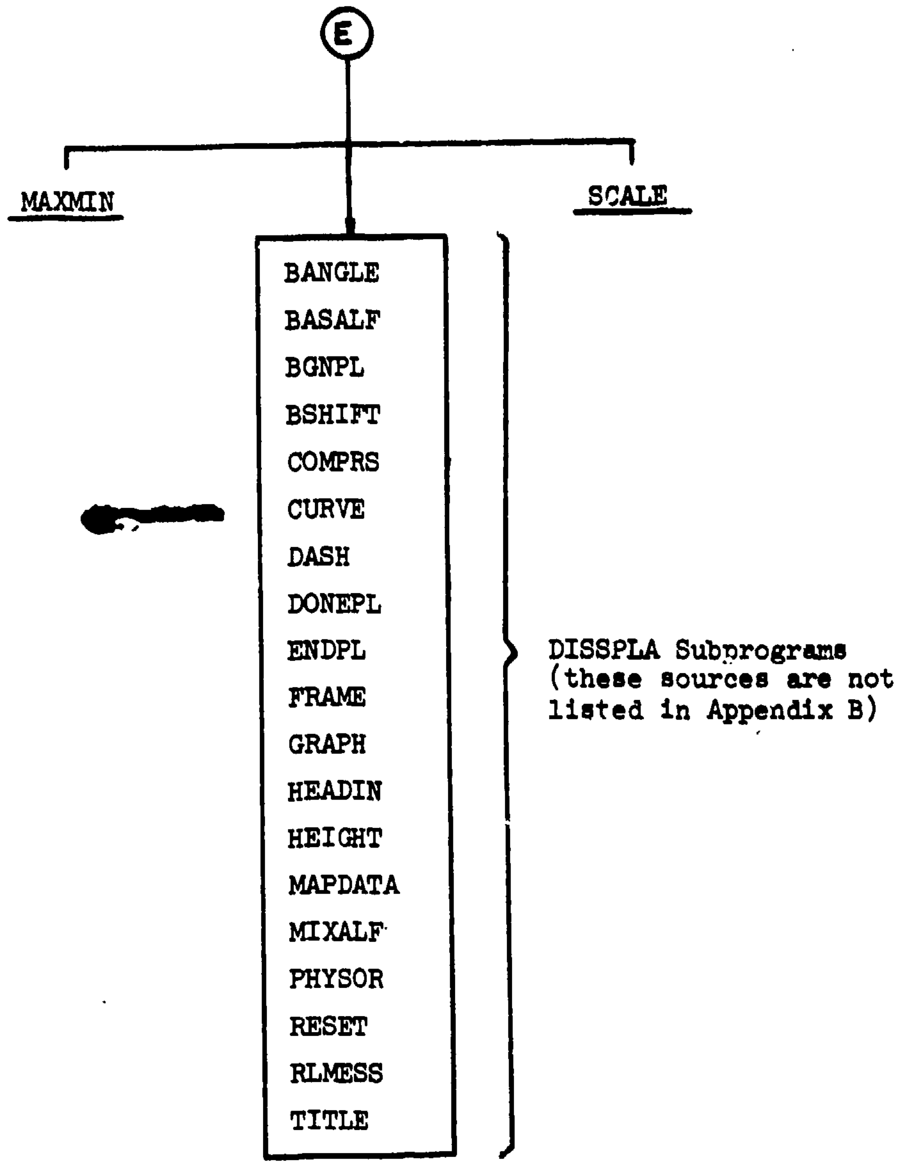


Figure 21 (Continued)

APPENDIX A

SAMPLE RUN OF PROFGEN

The sample run described here was constructed in seventeen segments to exercise most of PROFGEN's code including at least two segments of each of the four types of maneuvers. Throughout the sample run the nominal flight path is a great circle, the output interval is one second, and the integration step-size is variable. Figures 2 and 3 are the PRDATA and PASDATA lists that were used as input.

• Printed Output

Figure A-1 is a portion of the printed output from the sample run. The first page of printed output consists of a banner (automatically printed by the CYBER-74 computer) followed by the date and clock time of the run. The second and third pages are listings of the PRDATA and PASDATA lists as read from TAPE9, the local file on which the input should reside. These listings simply echo the data, including its mistakes if any.

Page 4 of Figure A-1 begins the printed output generated during the computational portion of the run with IPRNT=1. This output consists of a header and a list of variable values at the start of each segment, followed by output at DTO intervals (one second in this run) during the segment. The list of variables printed does not change

and the definition for each such variable, with its units, is given in Table A-1. Pages 5, 6 and 7 of Figure A-1 show output up through the beginning of segment 2.

The last page of the sample printout contains, in addition to output spaced at DTO intervals, output at t-final (460.5 seconds in this run) plus a post-run assessment of the numerical integration burden. In this case 5393 numerical integration steps were used and F was called 34675 times.

- Plotted Output

Figures A-2 through A-6 are the plotted output for the sample run. The small numbers appearing along the curve in each figure are segment numbers designating approximately where each new segment began. The latitude - longitude plot in Figure A-2 is constructed with the latitude and longitude axes at the same scale.

- Other Output

TAPE3 output was suppressed in the sample run by setting IRITE=0. If TAPE3 output had been specified (unformatted binary records), each record would have contained the following list of variables in units of feet, seconds and/or radians: time, latitude, longitude, alpha, altitude, roll, pitch, yaw, velocity components along nav x, y, z and specific force components along nav x, y, z. Subroutine RITEOUT should be consulted if a more definitive description of TAPE3 output is needed.

Table A-1 - Output Variables

Variable	Units	Description
TIME	sec.	time (t)
LAT	deg.	geographic latitude (ϕ)
LON	deg.	longitude (λ)
ALPHA	deg.	angle between north and nav X-axis (α)
ALT	feet	altitude from ellipsoid (h)
ROLL	deg.	roll (η_x)
PITCH	deg.	pitch (η_y)
YAW	deg.	yaw (η_z)
PSI	deg.	ground heading angle measured positive cw from north (ψ)
DROLL	deg/sec	derivative of roll ($\dot{\eta}_x$)
DPITCH	deg/sec	derivative of pitch ($\dot{\eta}_y$)
DYAW	deg/sec	derivative of yaw ($\dot{\eta}_z$)
VX	ft/sec	velocity w.r.t. earth along nav x-axis (V_x)
VY	ft/sec	velocity w.r.t. earth along nav y-axis (V_y)
VZ	ft/sec	velocity w.r.t. earth along nav z-axis (V_z)
VPATH	ft/sec	magnitude of total velocity (V_T)
FX	ft/sec^2	specific force along nav X-axis (F_x)
FY	ft/sec^2	specific force along nav y-axis (F_y)
FZ	ft/sec^2	specific force along nav z-axis (F_z)
APATH	ft/sec^2	acceleration along path X-axis (i.e. along $\underline{V}$)

```
$$$$$$$$  $$$$$$   $$    $$  $      $  $$$$$$   $$$$$ $  $$$$$$$$
   $$    $$    $$  $$    $$  $$    $$ $$    $$ $$    $  $$
   $$    $$        $$    $$  $$$  $$$ $$    $$ $$   $$$ $$
   $$     $$$$$$   $$$$$$$$  $$$$$$$$ $$    $$ $$  $ $$ $$$$$$$
   $$          $$  $$    $$  $$ $$ $$ $$    $$ $$ $$ $$       $$
   $$          $$  $$    $$  $$    $$ $$    $$ $$ $  $$       $$
   $$          $$  $$    $$  $$    $$ $$    $$ $$$   $$       $$
   $$    $$    $$  $$    $$  $$    $$ $$    $$ $    $$  $$    $$
$$$$$$$$  $$$$$$   $$    $$  $$    $$  $$$$$$  $ $$$$$   $$$$$$
```

TODAY = 11/16/76

CLOCK = 15.51.32.

Figure A-1 Sample Output (1 of 8)

```
$PRDATA
IPRGB   = 650,
NSEGT   = 17,
LLMECH  = 2,
TSTART  = 0.0.
VTO     = .1E+04,
PHEADO  = .18F+03,
PPITCHO = 0.0,
ALFAO   = .45E+02,
LATO    = .39E+02,
LONO    = -.84E+02,
ALTO    = .3E+05,
IPRNT   = 1,
IRITE   = 0,
IPLOT   = 1,
ROLRATE = .25E+03,
$END
```


(2 of 8)

```
$PASDATA

SEGLNT  = .2E+02, .3E+02, .1E+02, .3E+02, .3E+02, .4E+02, .1E+02, .1E+02,          .6E+02, .1E+02, .1E+02, .1F+02, .5E+02,
          .1E+02, .405E+02, .5E+02, .4E+02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

RESTART = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

TURN    = 4, 3, 4, 3, 2, 2, 1, 2, 4, 2, 1, 4, 2, 4, 2, 4, 2, 4, 4, 4, 4, 4, 4,  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
          4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,

NPATH   = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2.
          2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

PACC    = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.1E+00, 0.0, 0.0, 0.0, .1E+00, 0.0,     0.0, 0.0, 0.0, 0.0, .1E+01, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

TACC    = 0.0, 0.0, 0.0, 0.0, .1E+01, .1E+01, .5E+00, .5E+01, 0.0, .5E+01,         .5E+00, 0.0, .4E+01, 0.0, .2E+01, 0.0,
          .2E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

HEAD    = 0.0, .2E+02, 0.0, -.2E+02, -.3E+02, .3E+02, 0.0, -.9E+02, 0.0,           -.9E+02, 0.0, 0.0, .365E+03, 0.0, -.135E+03,
          0.0, .135E+03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0,

PITCH   = 0.0, .36E+02, 0.0, .36E+02, 0.0, 0.0, .5E+01, 0.0, 0.0, 0.0, -.5E+01,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

DTO     = .1E+01, .1E+01, .1E+01, .1E+01, .1E+01, .1E+01, .1F+01, .1E+01,          .1E+01, .1E+01, .1F+01, .1F+01, .1E+01,
          .1E+01, .1E+01, .1E+01, .1E+01, .1E+09, .1F+09, .1E+09, .1E+09,          .1E+09, .1E+09, .1E+09, .1F+09, .1E+09,
          .1E+09, .1E+09, .1E+09, .1E+09, .1E+09, .1E+09, .1E+09, .1E+09,          .1E+09, .1F+09, .1E+09, .1F+09, .1E+09,
          .1E+09, .1E+09, .1E+09, .1E+09, .1E+09, .1E+09, .1E+09, .1E+09,          .1E+09, .1E+09, .1E+09,

MODE    = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
          1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

ERROR   = .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1F-05, .1E-05, .1E-05,          .1E-05, .1E-05, .1E-05, .1E-05, .1E-05,
          .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05,          .1E-05, .1E-05, .1F-05, .1F-05, .1F-05,
          .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05,          .1E-05, .1E-05, .1E-05, .1E-05, .1E-05,
          .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05, .1E-05,          .1E-05, .1F-05, .1E-05,

HMAX    = .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1E+05,          .1E+05, .1E+05, .1F+05, .1E+05, .1E+05,
          .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1F+05, .1F+05,          .1F+05, .1E+05, .1E+05, .1F+05, .1E+05,
          .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1E+05,          .1E+05, .1E+05, .1E+05, .1E+05, .1E+05,
          .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1E+05, .1F+05, .1E+05,          .1E+05, .1E+05, .1E+05,

HMIN    = .1E-03, .1E-03, .1E-03, .1E-03, .1E-03, .1E-03, .1E-03, .1F-03,          .1E-03, .1F-03, .1E-03, .1E-03, .1E-03,
          .1E-03, .1E-03, .1E-03, .1E-03, .1E+01, .1F+01, .1E+01, .1F+01,          .1E+01, .1E+01, .1F+01, .1E+01, .1E+01,
          .1E+01, .1E+01, .1E+01, .1E+01, .1E+01, .1E+01, .1F+01, .1E+01,          .1E+01, .1E+01, .1E+01, .1E+01, .1E+01,
          .1E+01, .1E+01, .1E+01, .1E+01, .1E+01, .1E+01, .1E+01, .1E+01,          .1E+01, .1F+01, .1E+01,

$END
```

(3 of 8)

```
BEGIN SEGMENT NUMBER  1
INITIAL TIME        0.00000
FINAL TIME         20.00000
THIS FLIGHT SEGMENT IS A STRAIGHT FLIGHT.
THE NOMINAL FLIGHT PATH OVER THE EARTH IS A GREAT CIRCLE.
THE INTEGRATION STEP SIZE IS VARIABLE.
THE LOCAL LEVEL MECHANIZATION IS CONSTANT ALPHA.

TIME      0.00000
    LAT     39.00000000      LON     -84.00000000     ALPHA    45.00000000       ALT     30000.00000
    ROLL    0.               PITCH    0.              YAW     -135.0000000       PSI     180.0000000
    DROLL   0.               DPITCH   0.              DYAW     .1452136080E-15
    VX     -707.1067812      VY       707.1067812     VZ       0.                VPATH   1000.000000
    FX     -.6473823627E-01  FY      -.6506845547E-01 FZ       32.01467768       APATH   0.

TIME      1.00000
    LAT     38.99725638      LON     -84.00000000     ALPHA    45.00000000       ALT     30000.00000
    ROLL    0.               PITCH    0.              YAW     -135.0000000       PSI     180.0000000
    DROLL   0.               DPITCH   0.              DYAW     .6408621741E-16
    VX     -707.1067812      VY       707.1067812     VZ       0.                VPATH   1000.000000
    FX     -.6472640472E-01  FY      -.6506461703E-01 FZ       32.01466970       APATH   0.

TIME      2.00000
    LAT     38.99451675      LON     -84.00000000     ALPHA    45.00000000       ALT     30000.00000
    ROLL    0.               PITCH    0.              YAW     -135.0000000       PSI    -180.0000000
    DROLL   0.               DPITCH   0.              DYAW     .4005008575E-17
    VX     -707.1067812      VY       707.1067812     VZ       0.                VPATH   1000.000000
    FX     -.6472257382E-01  FY      -.6506077845E-01 FZ       32.01466171       APATH   0.

TIME      3.00000
    LAT     38.99177513      LON     -84.00000000     ALPHA    45.00000000       ALT     30000.00000
    ROLL    0.               PITCH    0.              YAW     -135.0000000       PSI    -180.0000000
    DROLL   0.               DPITCH   0.              DYAW    -.4004528769E-17
    VX     -707.1067812      VY       707.1067812     VZ       0.                VPATH   1000.000000
    FX     -.6471874117E-01  FY      -.6505693971E-01 FZ       32.01465373       APATH   0.

TIME      4.00000
    LAT     38.98903358      LON     -84.00000000     ALPHA    45.00000000       ALT     30000.00000
    ROLL    0.               PITCH    0.              YAW     -135.0000000       PSI    -180.0000000
    DROLL   0.               DPITCH   0.              DYAW     .4369568301E-22
    VX     -707.1067812      VY       707.1067812     VZ       0.                VPATH   1000.000000
    FX     -.6471490917E-01  FY      -.6505310082E-01 FZ       32.01464574       APATH   0.

TIME      5.00000
    LAT     38.98629197      LON     -84.00000000     ALPHA    45.00000000       ALT     30000.00000
    ROLL    0.               PITCH    0.              YAW     -135.0000000       PSI    -180.0000000
    DROLL   0.               DPITCH   0.              DYAW    -.4003743884E-17
    VX     -707.1067812      VY       707.1067812     VZ       0.                VPATH   1000.000000
    FX     -.6471107792E-01  FY      -.6504926178E-01 FZ       32.01463776       APATH   0.

TIME      6.00000
    LAT     38.98355024      LON     -84.00000000     ALPHA    45.00000000       ALT     30000.00000
    ROLL    0.               PITCH    0.              YAW     -135.0000000       PSI    -180.0000000
    DROLL   0.               DPITCH   0.              DYAW     .4003430951E-17
    VX     -707.1067812      VY       707.1067812     VZ       0.                VPATH   1000.000000
    FX     -.6470724472E-01  FY      -.6504542258E-01 FZ       32.01462978       APATH   0.
```

(4 of 8)

```
TIME      7.00000
      LAT      38.98080861         LON     -84.00000000       ALPHA     45.00000000        ALT      30000.00000
      ROLL     0.                  PITCH    0.                YAW      -135.0000000        PSI     -180.0000000
      DROLL    0.                  DPITCH   0.                DYAW     -.4002959119E-17
      VX      -707.1067812         VY       707.1067812       VZ        0.                 VPATH    1000.000000
      FX      -.6470341227E-01     FY      -.6504158324E-01   FZ        32.01462179        APATH    0.

TIME      8.00000
      LAT      38.97806698         LON     -84.00000000       ALPHA     45.00000000        ALT      30000.00000
      ROLL     0.                  PITCH    0.                YAW      -135.0000000        PSI     -180.0000000
      DROLL    0.                  DPITCH   0.                DYAW      .4376932623E-22
      VX      -707.1067812         VY       707.1067812       VZ        0.                 VPATH    1000.000000
      FX      -.6469957958E-01     FY      -.6503774374E-01   FZ        32.01461381        APATH    0.

TIME      9.00000
      LAT      38.97532535         LON     -84.00000000       ALPHA     45.00000000        ALT      30000.00000
      ROLL     0.                  PITCH    0.                YAW      -135.0000000        PSI     -180.0000000
      DROLL    0.                  DPITCH   0.                DYAW     -.4002174476E-17
      VX      -707.1067812         VY       707.1067812       VZ        0.                 VPATH    1000.000000
      FX      -.6469574694E-01     FY      -.6503390409E-01   FZ        32.01460582        APATH    0.

TIME     10.00000
      LAT      38.97258371         LON     -84.00000000       ALPHA     45.00000000        ALT      30000.00000
      ROLL     0.                  PITCH    0.                YAW      -135.0000000        PSI     -180.0000000
      DROLL    0.                  DPITCH   0.                DYAW      .4380613829E-22
      VX      -707.1067812         VY       707.1067812       VZ        0.                 VPATH    1000.000000
      FX      -.6469191494E-01     FY      -.6503006429E-01   FZ        32.01459784        APATH    0.

TIME     11.00000
      LAT      38.96984207         LON     -84.00000000       ALPHA     45.00000000        ALT      30000.00000
      ROLL     0.                  PITCH    0.                YAW      -135.0000000        PSI     -180.0000000
      DROLL    0.                  DPITCH   0.                DYAW     -.4001389354E-17
      VX      -707.1067812         VY       707.1067812       VZ        0.                 VPATH    1000.000000
      FX      -.6468808190E-01     FY      -.6502622433E-01   FZ        32.01458986        APATH    0.

TIME     12.00000
      LAT      38.96710044         LON     -84.00000000       ALPHA     45.00000000        ALT      30000.00000
      ROLL     0.                  PITCH    0.                YAW      -135.0000000        PSI     -180.0000000
      DROLL    0.                  DPITCH   0.                DYAW      .4384294397E-22
      VX      -707.1067812         VY       707.1067812       VZ        0.                 VPATH    1000.000000
      FX      -.6468424781E-01     FY      -.6502238423E-01   FZ        32.01458188        APATH    0.

TIME     13.00000
      LAT      38.96435880         LON     -84.00000000       ALPHA     45.00000000        ALT      30000.00000
      ROLL     0.                  PITCH    0.                YAW      -135.0000000        PSI     -180.0000000
      DROLL    0.                  DPITCH   0.                DYAW     -.4000605553E-17
      VX      -707.1067812         VY       707.1067812       VZ        0.                 VPATH    1000.000000
      FX      -.6468041448E-01     FY      -.6501854397E-01   FZ        32.01457389        APATH    0.

TIME     14.00000
      LAT      38.96161716         LON     -84.00000000       ALPHA     45.00000000        ALT      30000.00000
      ROLL     0.                  PITCH    0.                YAW      -135.0000000        PSI     -180.0000000
      DROLL    0.                  DPITCH   0.                DYAW      .4000301157E-17
      VX      -707.1067812         VY       707.1067812       VZ        0.                 VPATH    1000.000000
      FX      -.6467658099E-01     FY      -.6501470356E-01   FZ        32.01456591        APATH    0.
```

(5 of 8)

```
TIME      15.00000
     LAT       38.95887552     LON      -84.00000000     ALPHA     45.00000000      ALT     30000.00000
     ROLL      0.              PITCH    0.               YAW       -135.0000000     PSI     -180.0000000
     DROLL     0.              DPITCH   0.               DYAW      -.3999821273E-17
     VX        -707.1067812    VY       707.1067812      VZ        0.               VPATH   1000.000000
     FX        -.6467274735E-01 FY      -.6501086300E-01 FZ        32.01455793      APATH   0.

TIME      16.00000
     LAT       38.95613397     LON      -84.00000000     ALPHA     45.00000000      ALT     30000.00000
     ROLL      0.              PITCH    0.               YAW       -135.0000000     PSI     -180.0000000
     DROLL     0.              DPITCH   0.               DYAW      .4391653618E-22
     VX        -707.1067812    VY       707.1067812      VZ        0.               VPATH   1000.000000
     FX        -.6466891357E-01 FY      -.6500702229E-01 FZ        32.01454995      APATH   0.

TIME      17.00000
     LAT       38.95339223     LON      -84.00000000     ALPHA     45.00000000      ALT     30000.00000
     ROLL      0.              PITCH    0.               YAW       -135.0000000     PSI     -180.0000000
     DROLL     0.              DPITCH   0.               DYAW      -.3999037114E-17
     VX        -707.1067812    VY       707.1067812      VZ        0.               VPATH   1000.000000
     FX        -.6466507963E-01 FY      -.6500318142E-01 FZ        32.01454196      APATH   0.

TIME      18.00000
     LAT       38.95065059     LON      -84.00000000     ALPHA     45.00000000      ALT     30000.00000
     ROLL      0.              PITCH    0.               YAW       -135.0000000     PSI     -180.0000000
     DROLL     0.              DPITCH   0.               DYAW      .4395332278E-22
     VX        -707.1067812    VY       707.1067812      VZ        0.               VPATH   1000.000000
     FX        -.6466124555E-01 FY      -.6499934041E-01 FZ        32.01453398      APATH   0.

TIME      19.00000
     LAT       38.94790894     LON      -84.00000000     ALPHA     45.00000000      ALT     30000.00000
     ROLL      0.              PITCH    0.               YAW       -135.0000000     PSI     -180.0000000
     DROLL     0.              DPITCH   0.               DYAW      -.7996550123E-17
     VX        -707.1067812    VY       707.1067812      VZ        0.               VPATH   1000.000000
     FX        -.6465741132E-01 FY      -.6499549924E-01 FZ        32.01452600      APATH   0.

TIME      20.00000
     LAT       38.94516729     LON      -84.00000000     ALPHA     45.00000000      ALT     30000.00000
     ROLL      0.              PITCH    0.               YAW       -135.0000000     PSI     -180.0000000
     DROLL     0.              DPITCH   0.               DYAW      .4399010282E-22
     VX        -707.1067812    VY       707.1067812      VZ        0.               VPATH   1000.000000
     FX        -.6465357694E-01 FY      -.6499165792E-01 FZ        32.01451802      APATH   0.
```

BEGIN SEGMENT NUMBER 2
INITIAL TIME 20.00000
FINAL TIME 50.00000
THIS FLIGHT SEGMENT IS A SINE HEADING CHANGE
THE NOMINAL FLIGHT PATH OVER THE EARTH IS A GREAT CIRCLE.
THE INTEGRATION STEP SIZE IS VARIABLE.
THE LOCAL LEVEL MECHANIZATION IS CONSTANT ALPHA.

TIME 20.00000

LAT	38.94516729	LON	-84.00000000	ALPHA	45.00000000	ALT	30000.00000		
ROLL	0.	PITCH	0.	YAW	-135.0000000	PSI	-180.0000000		
DROLL	250.0000000	DPITCH	-.8142219985E-12	DYAW	.4399010282E-22				
VX	-707.1067812	VY	707.1067812	VZ	0.	VPATH	1000.000000		
FX	-.6465357694E-01	FY	-.6499165792E-01	FZ	32.01451802	APATH	0.		

TIME 21.00000

LAT	38.94242647	LON	-84.00006749	ALPHA	45.00000000	ALT	30000.00000
ROLL	65.52537060	PITCH	-.5679838075E-15	YAW	-132.1729768	PSI	-177.1729768
DROLL	3.527304764	DPITCH	-.6361109363E-14	DYAW	4.052963114		
VX	-671.3711187	VY	741.1213268	VZ	-.3846229019E-12	VPATH	1000.000000
FX	52.35874746	FY	47.43871184	FZ	32.02010512	APATH	0.

TIME 22.00000

LAT	38.93969519	LON	-84.00036422	ALPHA	45.00000000	ALT	30000.00000
ROLL	60.27669862	PITCH	.6310426594E-15	YAW	-128.2503941	PSI	-173.2503941
DROLL	-23.29905581	DPITCH	.5088887490E-13	DYAW	3.231183693		
VX	-619.0993512	VY	785.3126723	VZ	-.8276927869E-12	VPATH	1000.000000
FX	44.21928636	FY	34.85986547	FZ	32.02783712	APATH	0.

116

TIME 23.00000

LAT	38.93697620	LON	-84.00081429	ALPHA	45.00000000	ALT	30000.00000
ROLL	-60.27636593	PITCH	-.9179019857E-14	YAW	-128.2506633	PSI	-173.2506633
DROLL	-23.29905581	DPITCH	.5088887490E-13	DYAW	-3.231656498		
VX	-619.1030407	VY	785.3097638	VZ	-.2425667732E-10	VPATH	[illegible]08.000000
FX	-44.36211925	FY	-34.97340979	FZ	32.02782918	APATH	0.

TIME 24.00000

LAT	38.93424491	LON	-84.00111106	ALPHA	45.00000000	ALT	30000.00000
ROLL	-65.52503709	PITCH	-.9154993794E-14	YAW	-132.1735580	PSI	-177.1735580
DROLL	3.527304764	DPITCH	.1908332809E-13	DYAW	-4.053117275		
VX	-671.3786372	VY	741.1145158	VZ	-.2168967292E-10	VPATH	1000.000000
FX	-52.49296791	FY	-47.55391407	FZ	32.02008880	APATH	0.

TIME 25.00000

LAT	38.93150408	LON	-84.00117841	ALPHA	45.00000000	ALT	30000.00000
ROLL	.1361560541E-03	PITCH	.1024889322E-13	YAW	-135.0006262	PSI	179.9993738
DROLL	250.0000000	DPITCH	-.1628443997E-11	DYAW	.5852232868E-05		
VX	-707.1145088	VY	707.0990535	VZ	-.5405148183E-11	VPATH	1000.000000
FX	-.6457170448E-01	FY	-.6491116588E-01	FZ	32.01447700	APATH	0.

(7 of 8)

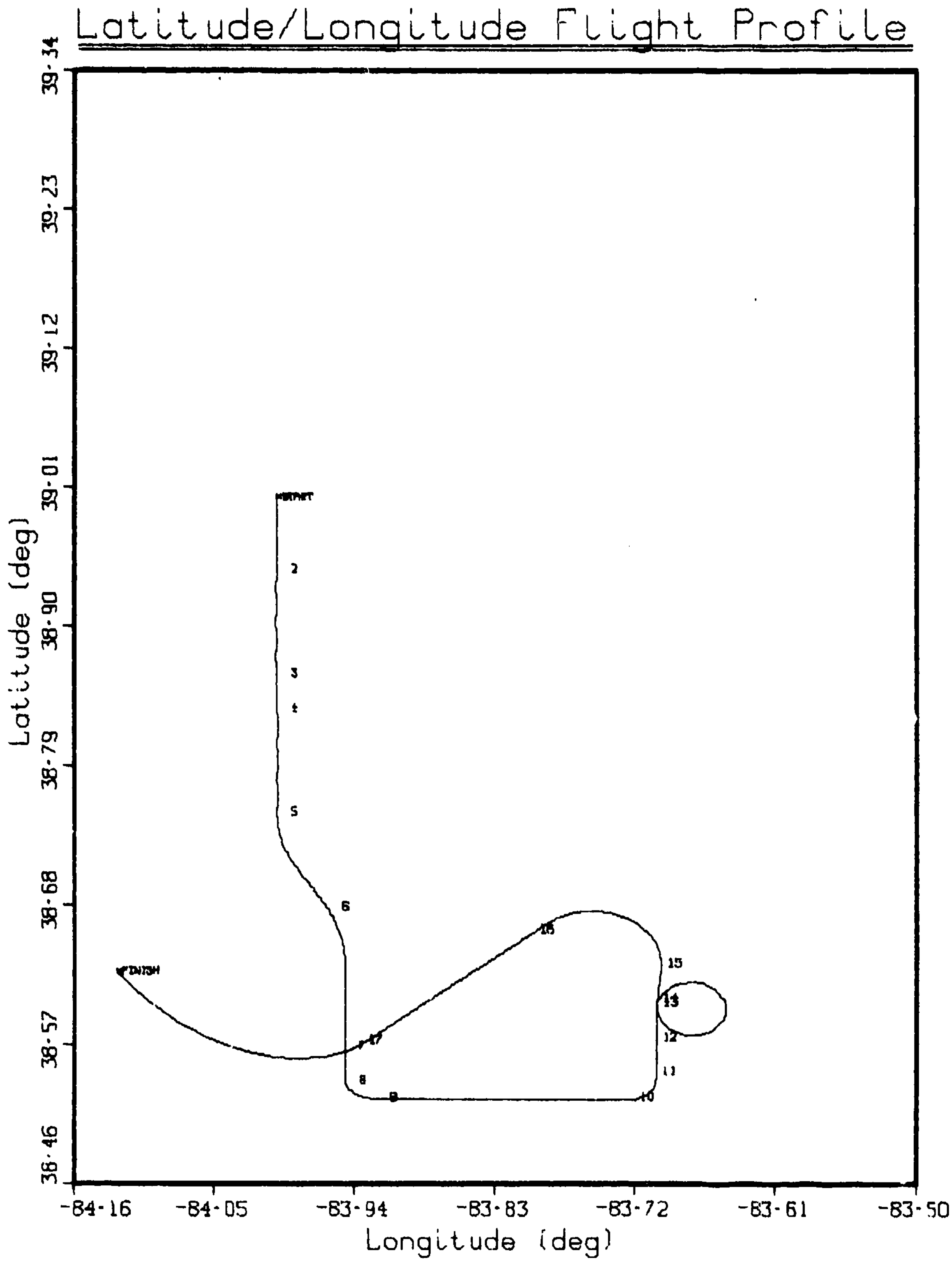
TIME 26.00000

LAT	38.92876325	LON	-84.00111099	ALPHA	45.00000000	ALT	[illegible]000.00000
ROLL	-65.52521476	PITCH	-.1156213180E-14	YAW	-137.8276305	PSI	[illegible]7.1723695
DROLL	-3.527304764	DPITCH	.6361109363E-14	DYAW	-4.052933776		
VX	-741.1284431	VY	671.3632630	VZ	-.2166464198E-10	VPATH	1000.000000
FX	-47.55287912	FY	-52.49472520	FZ	32.00887385	APATH	0.

```
TIME     456.00000
     LAT      38.60555833     LON     -84.10503992          ALPHA    45.00000000           ALT      37596.11896
     ROLL     63.43497896     PITCH   -.2995299753E-09      YAW      1.690553102           PSI      -43.38944690
     DROLL    0.              DPITCH   0.                   DYAW     1.718545553
     VX       2142.167190     VY      -63.22452895          VZ       -.1062224105E-07      VPATH    2143.100000
     FX       30.29203974     FY      -65.12776636          FZ       31.98631233           APATH    32.20000000

TIME     457.00000
     LAT      38.60992467     LON     -84.11013854          ALPHA    45.00000000           ALT      37596.11896
     ROLL     63.43497896     PITCH   -.2995299753E-09      YAW      3.395319533           PSI      -41.60368047
     DROLL    0.              DPITCH   0.                   DYAW     1.693114726
     VX       2171.479380     VY      -128.8696668          VZ       -.1079057571E-07      VPATH    2175.300000
     FX       28.34014960     FY      -65.99765320          FZ       31.97672065           APATH    32.20000000

TIME     458.00000
     LAT      38.61447397     LON     -84.11512826          ALPHA    45.00000000           ALT      37596.11896
     ROLL     63.43497896     PITCH   -.2995299753E-09      YAW      5.077031796           PSI      -39.92296820
     DROLL    0.              DPITCH   0.                   DYAW     1.668431693
     VX       2198.839140     VY      -195.3527255          VZ       -.1095891836E-07      VPATH    2207.500000
     FX       26.39252992     FY      -66.79752279          FZ       31.96680459           APATH    32.20000000

TIME     459.00000
     LAT      38.61920410     LON     -84.12002629          ALPHA    45.00000000           ALT      37596.11896
     ROLL     63.43497896     PITCH   -.2995299753E-09      YAW      6.733421439           PSI      -38.26657856
     DROLL    0.              DPITCH   0.                   DYAW     1.644464277
     VX       2224.251501     VY      -262.6049317          VZ       -.1112724582E-07      VPATH    2239.700000
     FX       24.45104364     FY      -67.52955813          FZ       31.95657025           APATH    32.20000000

TIME     460.00000
     LAT      38.62411243     LON     -84.12481801          ALPHA    45.00000000           ALT      37596.11896
     ROLL     63.43497896     PITCH   -.2995299753E-09      YAW      8.366188750           PSI      -36.63381125
     DROLL    0.              DPITCH   0.                   DYAW     1.621182117
     VX       2247.723274     VY      -330.5596689          VZ       -.1129557967E-07      VPATH    2271.900000
     FX       22.51741198     FY      -68.19589925          FZ       31.94602370           APATH    32.20000000

TIME     460.50000
     LAT      38.62663115     LON     -84.12717484          ALPHA    45.00000000           ALT      37596.11896
     ROLL     .3029935769E-04 PITCH   -.2995355128E-09      YAW      8.913073484           PSI      -36.00692692
     DROLL    -250.0000000    DPITCH   0.                   DYAW     -.2927341102E-02
     VX       2260.371345     VY      -354.4931340          VZ       -.2009382221E-06      VPATH    2288.000000
     FX       31.84358041     FY      -4.783832879          FZ       31.94159996           APATH    32.20000000

 THIS FLIGHT REQUIRED      5393    PASSES THROUGH THE NUMERICAL INTEGRATOR, KUTMER.
 KUTMER IN TURN MADE      34675    CALLS TO SUBROUTINE F, THE DERIVATIVE SUBPROGRAM.

***********        ISHM005   //// END  OF LIST ////
```

(8 of 8)

Figure A-2



Latitude/Longitude Flight Profile
Latitude (deg)
39·34
39·23
39·12
39·01
38·90
38·79
38·68
38·57
38·46
Longitude (deg)
-84·16
-84·05
-83·94
-83·83
-83·72
-83·61
-83·50
2
3
5
6
8
10
11
12
13
15
16
17

Figure A-3

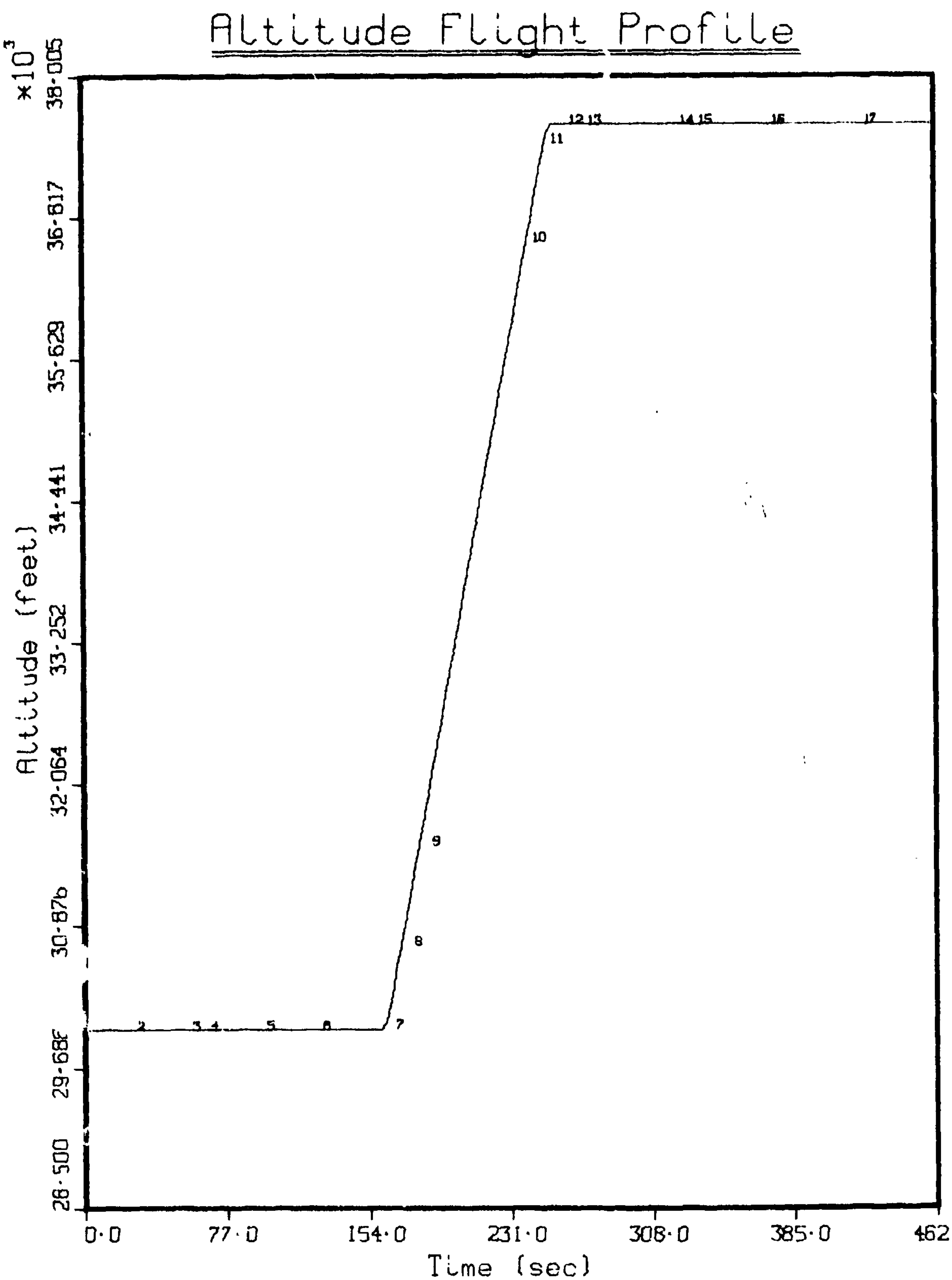
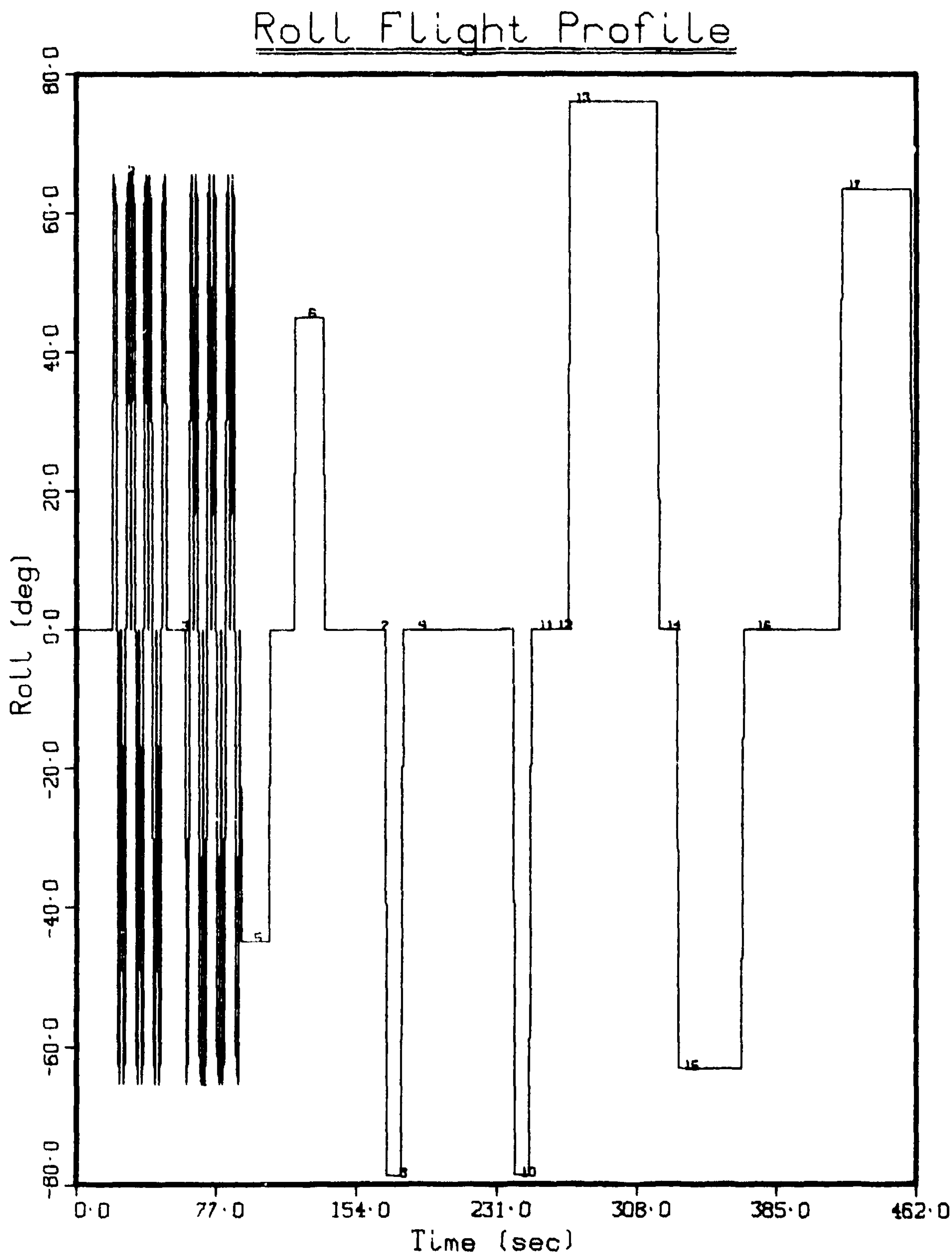

Altitude Flight Profile
*10^3
38.005
36.817
35.629
34.441
33.252
32.064
30.876
29.688
28.500
Altitude (feet)
0.0
77.0
154.0
231.0
308.0
385.0
462.
Time (sec)

Figure A-4



Roll Flight Profile
Roll (deg)
Time (sec)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
-80.0
0.0
77.0
154.0
231.0
308.0
385.0
462.0

Figure A-5

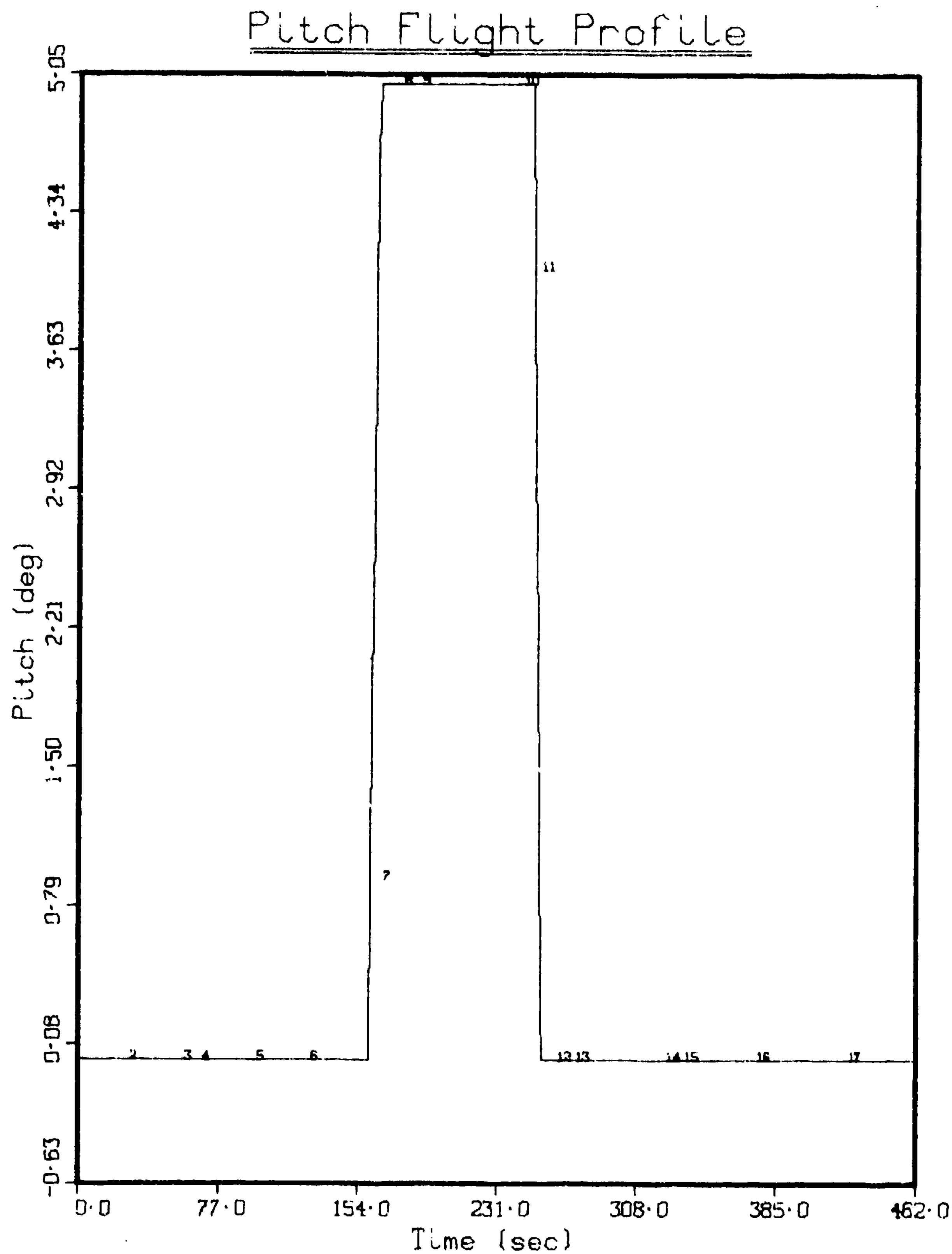
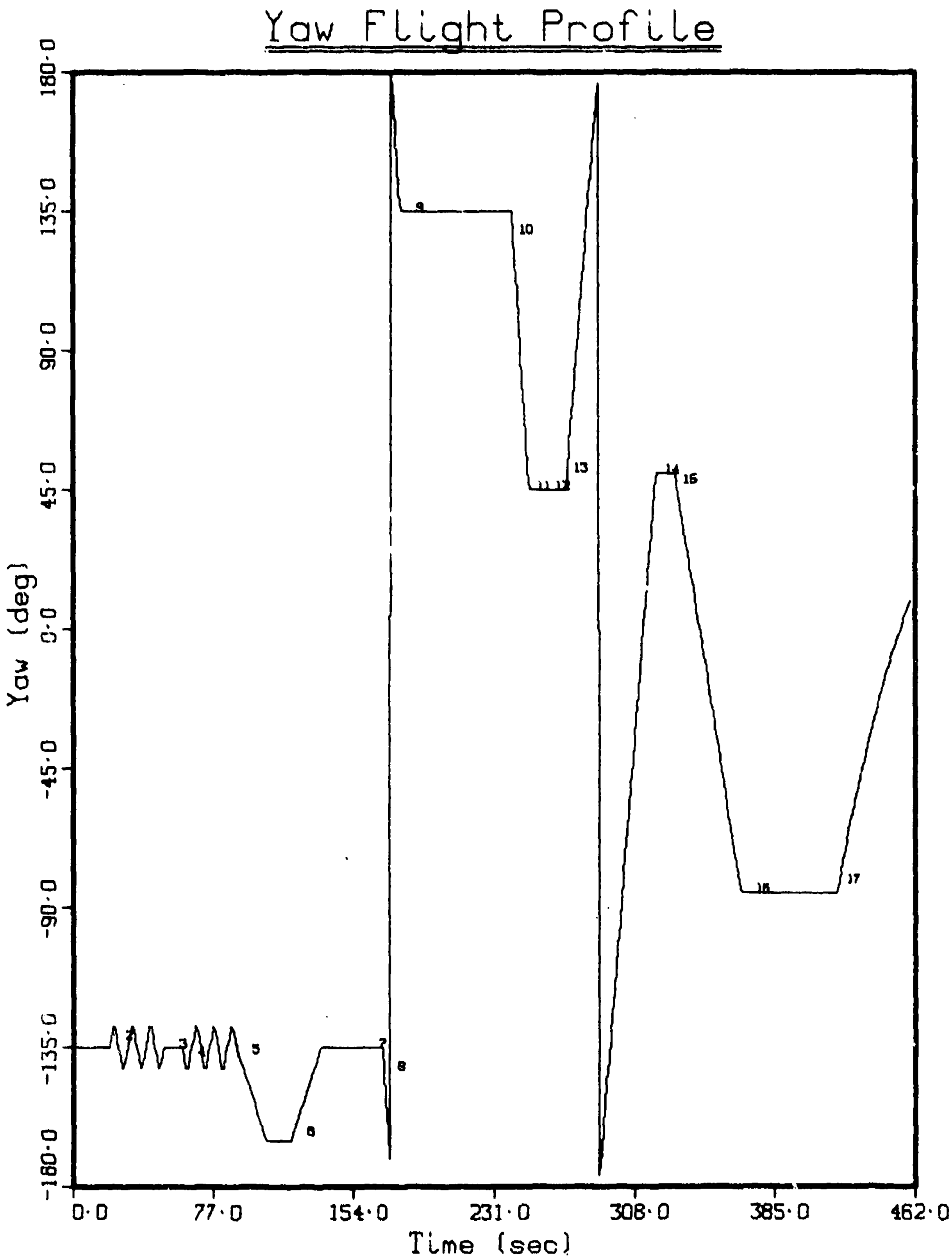

Pitch Flight Profile
Pitch (deg)
5·05
4·34
3·63
2·92
2·21
1·50
0·79
0·08
-0·63
0·0
77·0
154·0
231·0
308·0
385·0
462·0
Time (sec)
2
3
4
5
6
7
10
11
12
13
14
15
16
17

Figure A-6


Yaw Flight Profile
180.0
135.0
90.0
45.0
0.0
-45.0
-90.0
-135.0
-180.0
Yaw (deg)
0.0
77.0
154.0
231.0
308.0
385.0
462.0
Time (sec)

APPENDIX B

PROFGEN LISTING

```
      PROGRAM PROFGEN (INPUT,OUTPUT,TAPE3,TAPE6,TAPE9,                  PROFGEN    2
     1 PLFILE=0,MAPDTA=0)                                               PROFGEN    3
                                                                        PROFGEN    4
C**   THIS IS THE MAIN PROGRAM OF A SIMULATION DESIGNED TO GENERATE     PROFGEN    5
C**   FLIGHT PROFILES. A PROFILE CONSISTS OF A SEQUENCE OF UP TO 50     PROFGEN    6
C**   FLIGHT SEGMENTS, ONE FOLLOWING ANOTHER. EACH FLIGHT SEGMENT       PROFGEN    7
C**   EXECUTES ONE "MANEUVER". THESE KINDS OF MANEUVERS ARE POSSIBLE:   PROFGEN    8
C**          * VERTICAL TURNS                                           PROFGEN    9
C**          * HORIZONTAL TURNS                                         PROFGEN   10
C**          * SINUSOIDAL HEADING CHANGES                               PROFGEN   11
C**          * STRAIGHT FLIGHTS OVER GREAT CIRCLE OR RHUMB LINE PATHS   PROFGEN   12
C**   INPUT IS MADE VIA TWO NAMELISTS, PRDATA AND PASDATA. THE AIRCRAFT PROFGEN   13
C**   POSITION, VELOCITY, ACCELERATION AND ORIENTATION ARE COMPUTED     PROFGEN   14
C**   USING THE KINEMATIC EQUATIONS APPROPRIATE FOR AN ELLIPSOIDAL EARTH PROFGEN  15
C**   AND A LOCAL-LEVEL, ALPHA-CONTROLLED MECHANIZATION. OUTPUT CAN BE  PROFGEN   16
C**   TAYLORED TO THE USER'S NEEDS. COMPLETE INFORMATION ABOUT PROFGEN, PROFGEN   17
C**   ITS CAPABILITIES, ITS LIMITATIONS AND THE MEANS FOR USING IT, IS  PROFGEN   18
C**   GIVEN IN "PROFGEN - A COMPUTER PROGRAM FOR GENERATING FLIGHT      PROFGEN   19
C**   PROFILES". CONTACT RWA, AFAL, W-P AFB, OH. (513)255-6843. 2/11/76 PROFGEN   20
                                                                        PROFGEN   21
      COMMON /DTO/DTO(50)       /ERROR/ERROR(50)    /FIXED/FIXED(15)    PROFGEN   22
      COMMON /HEAD/HEAD(50)     /HMAX/HMAX(50)      /HMIN/HMIN(50)      PROFGEN   23
      COMMON /MODE/MODE(50)     /NPATH/NPATH(50)    /PACC/PACC(50)      PROFGEN   24
      COMMON /PITCH/PITCH(50)   /PRBLK/PRBLK(13)                        PROFGEN   25
      COMMON /RESTART/RESTART(50)/SEGLNT/SEGLNT(50)  /SUPLE/SUPLE(9)    PROFGEN   26
      COMMON /TACC/TACC(50)     /TURN/TURN(50)                          PROFGEN   27
                                                                        PROFGEN   28
      EQUIVALENCE (FIXED(3),TWOPI)                                      PROFGEN   29
      EQUIVALENCE (FIXED(4),PI)                                         PROFGEN   30
      EQUIVALENCE (FIXED(5),HALFPI)                                     PROFGEN   31
      EQUIVALENCE (PRBLK(1),LLMECH)                                     PROFGEN   32
      EQUIVALENCE (PRBLK(2),TSTART)                                     PROFGEN   33
      EQUIVALENCE (PRBLK(3),VTO)                                        PROFGEN   34
      EQUIVALENCE (PRBLK(4),PHEADO)                                     PROFGEN   35
      EQUIVALENCE (PRBLK(5),PPITCHO)                                    PROFGEN   36
      EQUIVALENCE (PRBLK(6),ALFAO)                                      PROFGEN   37
      EQUIVALENCE (PRBLK(7),LATO)                                       PROFGEN   38
      EQUIVALENCE (PRBLK(8),LONO)                                       PROFGEN   39
      EQUIVALENCE (PRBLK(9),ALTO)                                       PROFGEN   40
      EQUIVALENCE (PRBLK(10),IPRNT)                                     PROFGEN   41
      EQUIVALENCE (PRBLK(11),IRITE)                                     PROFGEN   42
      EQUIVALENCE (PRBLK(12),IPLOT)                                     PROFGEN   43
      EQUIVALENCE (PRBLK(13),ROLRATE)                                   PROFGEN   44
      EQUIVALENCE (SUPLE(1),T)                                          PROFGEN   45
      EQUIVALENCE (SUPLE(2),TF)                                         PROFGEN   46
      EQUIVALENCE (SUPLE(3),TI)                                         PROFGEN   47
      EQUIVALENCE (SUPLE(6),ISEG)                                       PROFGEN   48
                                                                        PROFGEN   49
      INTEGER RESTART,TURN                                              PROFGEN   50
      REAL LATO,LONO                                                    PROFGEN   51
                                                                        PROFGEN   52
      NAMELIST /PRDATA/ IPROB,NSEGT,LLMECH,TSTART,VTO,PHEADO,PPITCHO,   PROFGEN   53
     1 ALFAO,LATO,LONO,ALTO,IPRNT,IRITE,IPLOT,ROLRATE                   PROFGEN   54
      NAMELIST /PASDATA/ SEGLNT,RESTART,TURN,NPATH,PACC,TACC,HEAD,      PROFGEN   55
     1 PITCH,DTO,MODE,ERROR,HMAX,HMIN                                   PROFGEN   56
                                                                        PROFGEN   57
C                                                                       PROFGEN   58
```

```
      C          PRINT DATE AND TIME                                PROFGEN   59
            TODAY=DATE(DMY)                                         PROFGEN   60
            CLOCK=TIME(DMY)                                         PROFGEN   61
            WRITE (6,100) TODAY,CLOCK                               PROFGEN   62
      C                                                             PROFGEN   63
      C          READ, PRINT AND VALIDATE INPUT DATA                PROFGEN   64
            READ (9,PRDATA)                                         PROFGEN   65
            READ (9,PASDATA)                                        PROFGEN   66
            WRITE (6,PRDATA)                                        PROFGEN   67
            WRITE (6,PASDATA)                                       PROFGEN   68
            CALL VALDATA(NSEGT)                                     PROFGEN   69
            IF (IRITE.NE.1) GO TO 10                                PROFGEN   70
      C                                                             PROFGEN   71
      C          WRITE INPUT DATA ON A TAPE ANNOTATED WITH DATE AND TIME  PROFGEN   72
            REWIND 3                                                PROFGEN   73
            WRITE (3) TODAY,CLOCK                                   PROFGEN   74
            WRITE (3) IPROB,NSEGT,LLMECH,TSTART,VTO,PHEADO,PPITCHO, PROFGEN   75
           1 ALFAO,LATO,LONO,ALTO,IPRNT,IRITE,IPLOT,ROLRATE          PROFGEN   76
            WRITE (3) SEGLNT,RESTART,TURN,NPATH,PACC,TACC,HEAD,     PROFGEN   77
           1 PITCH,DTO,MODE,ERROR,HMAX,HMIN                          PROFGEN   78
      C                                                             PROFGEN   79
      C          CONVERT INPUT TO UNITS OF FEET, SECONDS AND RADIANS PROFGEN   80
         10 CALL NEWUNIT                                            PROFGEN   81
      C                                                             PROFGEN   82
      C          COMPUTE MACHINE-CRITICAL CONSTANTS FOR             PROFGEN   83
      C          STORAGE IN COMMON BLOCK "FIXED"                    PROFGEN   84
            PI=ABS(ATAN2(0.,-1.))                                   PROFGEN   85
            HALFPI=PI/2.                                            PROFGEN   86
            TWOPI=2.*PI                                             PROFGEN   87
      C                                                             PROFGEN   88
      C          INITIALIZE TIME AND THEN ENTER LOOP                PROFGEN   89
      C          GOVERNING PASSAGE TO EACH FLIGHT SEGMENT           PROFGEN   90
            T=TSTART                                                PROFGEN   91
            DO 20 I=1,NSEGT                                         PROFGEN   92
               ISEG=I                                               PROFGEN   93
               IF (RESTART(I).EQ.1) T=TSTART                        PROFGEN   94
               TI=T                                                 PROFGEN   95
               TF=T+SEGLNT(I)                                       PROFGEN   96
               CALL HEADER                                          PROFGEN   97
               CALL FLTPATH                                         PROFGEN   98
         20 CONTINUE                                                PROFGEN   99
      C                                                             PROFGEN  100
      C          POST-FLIGHT OUTPUT                                 PROFGEN  101
            TI=T                                                    PROFGEN  102
            CALL PRNTOUT                                            PROFGEN  103
            CALL KMPERF                                             PROFGEN  104
            IF (IRITE.EQ.1) CALL RITEOUT                            PROFGEN  105
            IF (IPLOT.EQ.1)  CALL PLOTTER(NSEGT)                    PROFGEN  106
            STOP                                                    PROFGEN  107
        100 FORMAT(//////T2,*TODAY = *,A10//T2,*CLOCK = *,A10)      PROFGEN  108
            END                                                     PROFGEN  109
```

```
      SUBROUTINE FLTPATH

C**   FLTPATH CONTROLS THE FLIGHT PATH GENERATION PROCESS DURING
C**   EACH FLIGHT SEGMENT. THE PASSAGE FROM SEGMENT TO SEGMENT IS
C**   GOVERNED BY PROFGEN, THE MAIN PROGRAM.

      COMMON /ERROR/ERROR(50)
      COMMON /FIXED/FIXED(15)
      COMMON /HMAX/HMAX(50)
      COMMON /HMIN/HMIN(50)
      COMMON /MODE/MODE(50)
      COMMON /PRBLK/PRBLK(13)
      COMMON /RESTART/RESTART(50)
      COMMON /STATE/X(23)
      COMMON /SUPLE/SUPLE(9)
      COMMON /TURN/TURN(50)

      EQUIVALENCE (FIXED(1),N)
      EQUIVALENCE (PRBLK(2),TSTART)
      EQUIVALENCE (PRBLK(10),IPRNT)
      EQUIVALENCE (PRBLK(11),IRITE)
      EQUIVALENCE (PRBLK(12),IPLOT)
      EQUIVALENCE (SUPLE(1),T)
      EQUIVALENCE (SUPLE(2),TF)
      EQUIVALENCE (SUPLE(4),TRNDONE)
      EQUIVALENCE (SUPLE(5),TDONE)
      EQUIVALENCE (SUPLE(6),ISEG)
      EQUIVALENCE (SUPLE(7),TOFF)
      EQUIVALENCE (SUPLE(8),TON)
      EQUIVALENCE (SUPLE(9),RRCOEF)

      INTEGER RESTART,TURN
      EXTERNAL F
C
C       INITIALIZE THE STATE VECTOR PRIOR TO BEGINNING THE FIRST
C       SEGMENT OR WHENEVER A PROBLEM RESTART IS REQUIRED.
      IF (ISEG.EQ.1 .OR. RESTART(ISEG).EQ.1) CALL SVSETUP
C
C       INITIALIZE SEVERAL PARAMETERS THAT ARE FLIGHT SEGMENT DEPENDENT
      H=HMN=HMIN(ISEG)
      MDE=MODE(ISEG)
      ERR=ERROR(ISEG)
      HMX=HMAX(ISEG)
      TRNDONE=1.
      RRCOEF=0.
      IF (TURN(ISEG).EQ.1) CALL TSETUP1(TDONE)
      IF (TURN(ISEG).EQ.1) TRNDONE=0.
      IF (TURN(ISEG).EQ.2) CALL TSETUP2(TOFF,TON,TDONE)
      IF (TURN(ISEG).EQ.2) TRNDONE=0.
      IF (TURN(ISEG).EQ.2) RRCOEF=+1.
      IF (TURN(ISEG).EQ.3) CALL CHKSHC
      IF (TURN(ISEG).EQ.3) RRCOEF=+1.
C
C       PRINT THE VALUE OF EACH VARIABLE AT THE BEGINNING OF EACH
C       SEGMENT AND THEN TRANSFER CONTROL TO THE SPECIFIED MANEUVER
      CALL PRNTOUT
```

```
      IF (IRITE.EQ.1 .AND. T.EQ.TSTART) CALL RITEOUT                    FLTPATH   59
      IF (IPLOT.EQ.1) CALL ARAYFIL                                      FLTPATH   60
      GO TO (40,50,60,70), TURN(ISEG)                                   FLTPATH   61
C                                                                       FLTPATH   62
C                                                                       FLTPATH   63
C         ***   VERTICAL TURN   ***                                     FLTPATH   64
C                                                                       FLTPATH   65
C                                                                       FLTPATH   66
   40 H=HLIMIT(T,TF,H,HMN)                                              FLTPATH   67
      H=HCHOP(H,T,TDONE)                                                FLTPATH   68
      CALL KUTMER(N,T,X,H,F,MDE,ERR,HMX,HMN)                            FLTPATH   69
      IF (T.GE.TDONE) TRNDONE=1.                                        FLTPATH   70
      IF (IPLOT.EQ.1) CALL ARAYFIL                                      FLTPATH   71
      IF (IPRNT.EQ.1) CALL PRNTOUT                                      FLTPATH   72
      IF (IRITE.EQ.1) CALL RITEOUT                                      FLTPATH   73
      IF (T.LT.TF) GO TO 40                                             FLTPATH   74
      RETURN                                                            FLTPATH   75
C                                                                       FLTPATH   76
C                                                                       FLTPATH   77
C         ***   HORIZONTAL TURN   ***                                   FLTPATH   78
C                                                                       FLTPATH   79
C                                                                       FLTPATH   80
   50 H=HLIMIT(T,TF,H,HMN)                                              FLTPATH   81
      CALL HLIM2(H,RRCOEF)                                              FLTPATH   82
      CALL KUTMER(N,T,X,H,F,MDE,ERR,HMX,HMN)                            FLTPATH   83
      IF (T.GE.TDONE) TRNDONE=1.                                        FLTPATH   84
      IF (IPLOT.EQ.1) CALL ARAYFIL                                      FLTPATH   85
      IF (IPRNT.EQ.1) CALL PRNTOUT                                      FLTPATH   86
      IF (IRITE.EQ.1) CALL RITEOUT                                      FLTPATH   87
      IF (T.LT.TF) GO TO 50                                             FLTPATH   88
      RETURN                                                            FLTPATH   89
C                                                                       FLTPATH   90
C                                                                       FLTPATH   91
C         ***   SINUSOIDAL HEADING CHANGE   ***                         FLTPATH   92
C                                                                       FLTPATH   93
C                                                                       FLTPATH   94
   60 H=HLIMIT(T,TF,H,HMN)                                              FLTPATH   95
      CALL HLIM3(H,RRCOEF)                                              FLTPATH   96
      CALL KUTMER(N,T,X,H,F,MDE,ERR,HMX,HMN)                            FLTPATH   97
      IF (IPLOT.EQ.1) CALL ARAYFIL                                      FLTPATH   98
      IF (IPRNT.EQ.1) CALL PRNTOUT                                      FLTPATH   99
      IF (IRITE.EQ.1) CALL RITEOUT                                      FLTPATH  100
      IF (T.LT.TF) GO TO 60                                             FLTPATH  101
      RETURN                                                            FLTPATH  102
C                                                                       FLTPATH  103
C                                                                       FLTPATH  104
C         ***   STRAIGHT FLIGHT   ***                                   FLTPATH  105
C                                                                       FLTPATH  106
C                                                                       FLTPATH  107
   70 H=HLIMIT(T,TF,H,HMN)                                              FLTPATH  108
      CALL KUTMER(N,T,X,H,F,MDE,ERR,HMX,HMN)                            FLTPATH  109
      IF (IPLOT.EQ.1) CALL ARAYFIL                                      FLTPATH  110
      IF (IPRNT.EQ.1) CALL PRNTOUT                                      FLTPATH  111
      IF (IRITE.EQ.1) CALL RITEOUT                                      FLTPATH  112
      IF (T.LT.TF) GO TO 70                                             FLTPATH  113
      RETURN                                                            FLTPATH  114
      END                                                               FLTPATH  115
```

```
      SUBROUTINE ACCLRTN(FX,FY,FZ)                                      ACCLRTN    2
                                                                        ACCLRTN    3
C**   ACCLRTN COMPUTES SPECIFIC FORCE WHICH IS THE TOTAL INERTIAL       ACCLRTN    4
C**   ACCELERATION MINUS THE MASS-ATTRACTION GRAVITATIONAL ACCELERATION. ACCLRTN   5
C**   SPECIFIC FORCE IS THE ACCELERATION THAT AN ACCELEROMETER MEASURES. ACCLRTN   6
C**   THE SPECIFIC FORCE RESULTS ARE EXPRESSED IN NAV COORDINATES.      ACCLRTN    7
                                                                        ACCLRTN    8
      COMMON /FIXED/FIXED(15)                                           ACCLRTN    9
      COMMON /STATE/STATE(23)                                           ACCLRTN   10
                                                                        ACCLRTN   11
      EQUIVALENCE (FIXED(8),WEI)                                        ACCLRTN   12
      EQUIVALENCE (STATE( 1),VX)                                        ACCLRTN   13
      EQUIVALENCE (STATE( 2),VY)                                        ACCLRTN   14
      EQUIVALENCE (STATE( 3),VZ)                                        ACCLRTN   15
      EQUIVALENCE (STATE(15),CEN11)                                     ACCLRTN   16
      EQUIVALENCE (STATE(16),CEN21)                                     ACCLRTN   17
      EQUIVALENCE (STATE(17),CEN31)                                     ACCLRTN   18
      EQUIVALENCE (RHO(1),RHOX)                                         ACCLRTN   19
      EQUIVALENCE (RHO(2),RHOY)                                         ACCLRTN   20
      EQUIVALENCE (RHO(3),RHOZ)                                         ACCLRTN   21
                                                                        ACCLRTN   22
      DIMENSION RHO(3)                                                  ACCLRTN   23
                                                                        ACCLRTN   24
      CALL VDOT(VXDOT,VYDOT,VZDOT)                                      ACCLRTN   25
      CALL GRAVITY(GX,GY,GZ)                                            ACCLRTN   26
      CALL RHONE(RHO)                                                   ACCLRTN   27
      WEIX=WEI*CEN11                                                    ACCLRTN   28
      WEIY=WEI*CEN21                                                    ACCLRTN   29
      WEIZ=WEI*CEN31                                                    ACCLRTN   30
      FX=VXDOT+(RHOY+2.*WEIY)*VZ-(RHOZ+2.*WEIZ)*VY-GX                   ACCLRTN   31
      FY=VYDOT+(RHOZ+2.*WEIZ)*VX-(RHOX+2.*WEIX)*VZ-GY                   ACCLRTN   32
      FZ=VZDOT+(RHOX+2.*WEIX)*VY-(RHOY+2.*WEIY)*VX-GZ                   ACCLRTN   33
      RETURN                                                            ACCLRTN   34
      END                                                               ACCLRTN   35
```

```
          REAL FUNCTION ALFA(DMY)                          ALFA      2
                                                           ALFA      3
C**   ALFA COMPUTES WANDER ANGLE, ALPHA                    ALFA      4
                                                           ALFA      5
      COMMON /STATE/STATE(23)                              ALFA      6
      EQUIVALENCE (STATE(15),CEN11)                        ALFA      7
      EQUIVALENCE (STATE(16),CEN21)                        ALFA      8
                                                           ALFA      9
      ALFA=ATAN2(-CEN21,CEN11)                             ALFA     10
      RETURN                                               ALFA     11
      END                                                  ALFA     12
```

```
      REAL FUNCTION ALFADOT(LLMECH)                             ALFADOT   2
                                                                ALFADOT   3
C**   ALFADOT COMPUTES THE ANGULAR RATE-OF-CHANGE OF ALPHA      ALFADOT   4
                                                                ALFADOT   5
      COMMON /FIXED/FIXED(15)                                   ALFADOT   6
      COMMON /STATE/STATE(23)                                   ALFADOT   7
                                                                ALFADOT   8
      EQUIVALENCE (FIXED( 8),WEI)                               ALFADOT   9
      EQUIVALENCE (STATE(17),SINEPHI)                           ALFADOT  10
                                                                ALFADOT  11
      REAL J,LAMDOT                                             ALFADOT  12
                                                                ALFADOT  13
      GO TO (10,20,30,40) LLMECH                                ALFADOT  14
   10 ALFADOT=-LAMDOT(DMY)*SINEPHI                              ALFADOT  15
      RETURN                                                    ALFADOT  16
   20 ALFADOT=0.                                                ALFADOT  17
      RETURN                                                    ALFADOT  18
   30 J=SIGN(1.,PHI(DMY))                                       ALFADOT  19
      ALFADOT=-J*LAMDOT(DMY)                                    ALFADOT  20
      RETURN                                                    ALFADOT  21
   40 ALFADOT=-(WEI+LAMDOT(DMY))*SINEPHI                        ALFADOT  22
      RETURN                                                    ALFADOT  23
      END                                                       ALFADOT  24
```

```
           SUBROUTINE ARAYFIL                                              ARAYFIL     2
CC                                                                         ARAYFIL     3
CC          STORES DATA FOR POST-RUN PLOTTING EVERY DTO SECONDS            ARAYFIL     4
CC                                                                         ARAYFIL     5
           COMMON /FIXED/FIXED(15)                                         ARAYFIL     6
           COMMON /GLON/GLON(1001)                                         ARAYFIL     7
           COMMON /GLAT/GLAT(1001)                                         ARAYFIL     8
           COMMON /GTIM/GTIM(1001)                                         ARAYFIL     9
           COMMON /GALT/GALT(1001)                                         ARAYFIL    10
           COMMON /GETX/GETX(1001)                                         ARAYFIL    11
           COMMON /GETY/GETY(1001)                                         ARAYFIL    12
           COMMON /GETZ/GETZ(1001)                                         ARAYFIL    13
           COMMON /NPLOT/NPLTPTS,NPLTSEG(50)                               ARAYFIL    14
           COMMON /STATE/X(23)                                             ARAYFIL    15
           COMMON /SUPLE/SUPLE(9)                                          ARAYFIL    16
                                                                           ARAYFIL    17
           EQUIVALENCE (SUPLE(1),T)                                        ARAYFIL    18
           EQUIVALENCE (SUPLE(3),TI)                                       ARAYFIL    19
           EQUIVALENCE (SUPLE(5),ISEG)                                     ARAYFIL    20
           EQUIVALENCE (FIXED(2),RADPERD)                                  ARAYFIL    21
           EQUIVALENCE (X(5),ALT)                                          ARAYFIL    22
                                                                           ARAYFIL    23
           REAL LAMDA                                                      ARAYFIL    24
                                                                           ARAYFIL    25
           DATA I/0/,IFULL/0/                                              ARAYFIL    26
                                                                           ARAYFIL    27
           IF (IFULL.EQ.1) RETURN                                          ARAYFIL    28
           TOUTNEW = TOUT(DMY)                                             ARAYFIL    29
           IF (T.EQ.TI) GO TO 10                                           ARAYFIL    30
           IF (TOUTOLD.EQ.TOUTNEW) RETURN                                  ARAYFIL    31
           IF (I.EQ.1001) WRITE(6,1000)                                    ARAYFIL    32
           IF (I.EQ.1001) IFULL = 1                                        ARAYFIL    33
           IF (I.EQ.1001) RETURN                                           ARAYFIL    34
                                                                           ARAYFIL    35
10         I = I+1                                                         ARAYFIL    36
           NPLTPTS = I                                                     ARAYFIL    37
           IF (T.EQ.TI) NPLTSEG(ISEG) = I+1                                ARAYFIL    38
           GLON(I) = LAMDA(DMY)/RADPERD                                    ARAYFIL    39
           GLAT(I) = PHI(DMY)/RADPERD                                      ARAYFIL    40
           GTIM(I) =  T                                                    ARAYFIL    41
           GALT(I) = ALT                                                   ARAYFIL    42
           GETX(I) = ETAX(DMY)/RADPERD                                     ARAYFIL    43
           GETY(I) = ETAY(DMY)/RADPERD                                     ARAYFIL    44
           GETZ(I) = ETAZ(DMY)/RADPERD                                     ARAYFIL    45
           TOUTOLD = TOUTNEW                                               ARAYFIL    46
           RETURN                                                          ARAYFIL    47
1000       FORMAT(/T2,"****** W A R N I N G :  PLOT ARRAYS ARE FULL.  ONLY FIRARAYFIL    48
          1ST 1001 POINTS WILL APPEAR ON PLOT.")                           ARAYFIL    49
           END                                                             ARAYFIL    50
```

```
      SUBROUTINE AXB(A,B,R,M,K,N)                                       AXB    2
C                                                                       AXB    3
C     ..................................................................AXB    4
C                                                                       AXB    5
C        SUBROUTINE AXB                                                 AXB    6
C                                                                       AXB    7
C        PURPOSE                                                        AXB    8
C           MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT           AXB    9
C           GENERAL MATRIX                                              AXB   10
C                                                                       AXB   11
C        USAGE                                                          AXB   12
C           CALL AXB(A,B,R,M,K,N)                                       AXB   13
C                                                                       AXB   14
C        DESCRIPTION OF PARAMETERS                                      AXB   15
C           A - NAME OF FIRST INPUT MATRIX                              AXB   16
C           B - NAME OF SECOND INPUT MATRIX                             AXB   17
C           R - NAME OF OUTPUT MATRIX                                   AXB   18
C           M - NUMBER OF ROWS IN A AND R                               AXB   19
C           K - NUMBER OF COLUMNS IN A AND ROWS IN B                    AXB   20
C           N - NUMBER OF COLUMNS IN B AND R                            AXB   21
C                                                                       AXB   22
C        REMARKS                                                        AXB   23
C           MATRIX R CAN BE IN THE SAME LOCATION AS EITHER              AXB   24
C           MATRIX A OR MATRIX B.                                       AXB   25
C           NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER       AXB   26
C           OF ROWS OF MATRIX B.                                        AXB   27
C           MATRIX D IS USED FOR TEMPORARY STORAGE AND MUST HAVE        AXB   28
C           FIXED DIMENSIONS AT LEAST AS LARGE AS THOSE OF R.           AXB   29
C                                                                       AXB   30
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  AXB   31
C           NONE                                                        AXB   32
C                                                                       AXB   33
C     ..................................................................AXB   34
C                                                                       AXB   35
      DIMENSION A(M,K), B(K,N), R(M,N), D(3,3)                          AXB   36
      DO 10 I=1,M                                                       AXB   37
      DO 10 J=1,N                                                       AXB   38
      D(I,J)=0.0                                                        AXB   39
      DO 10 L=1,K                                                       AXB   40
   10 D(I,J)=D(I,J)+A(I,L)*B(L,J)                                       AXB   41
      DO 20 I=1,M                                                       AXB   42
      DO 20 J=1,N                                                       AXB   43
   20 R(I,J)=D(I,J)                                                     AXB   44
      RETURN                                                            AXB   45
      END                                                               AXB   46
```

```
      BLOCK DATA                                                      BLKDAT     2
                                                                      BLKDAT     3
C                                                                     BLKDAT     4
C**  SOME DEFINITIONS AND PROBLEM INFORMATION                         BLKDAT     5
C                                                                     BLKDAT     6
C       COMMON BLOCK "STATE" CONTAINS THE STATE VECTOR                BLKDAT     7
C          STATE(1)    AIRCRAFT VELOCITY W.R.T. EARTH ALONG NAV X-AXIS BLKDAT    8
C          STATE(2)    AIRCRAFT VELOCITY W.R.T. EARTH ALONG NAV Y-AXIS BLKDAT    9
C          STATE(3)    AIRCRAFT VELOCITY W.R.T. EARTH ALONG NAV Z-AXIS BLKDAT   10
C          STATE(4)    AIRCRAFT VELOCITY W.R.T. EARTH ALONG PATH X-AXIS BLKDAT  11
C          STATE(5)    AIRCRAFT ALTITUDE                              BLKDAT    12
C          STATE 6-14  ELEMENTS OF PATH TO NAV DIRECTION COSINE MATRIX BLKDAT   13
C          STATE 15-23 ELEMENTS OF EARTH TO NAV DIRECTION COSINE MATRIX BLKDAT  14
C             THESE TWO 3X3 MATRICES ARE STORED AS FOLLOWS:           BLKDAT    15
C                CPN=(6  9 12)     CEN=(15 18 21)                     BLKDAT    16
C                    (7 10 13)         (16 19 22)                     BLKDAT    17
C                    (8 11 14)         (17 20 23)                     BLKDAT    18
C                                                                     BLKDAT    19
C       COMMON BLOCK "PRBLK" CONTAINS DATA FROM PRDATA NAMELIST       BLKDAT    20
C          PRBLK(1)    LLMECH, LOCAL LEVEL MECHANIZATION INDEX        BLKDAT    21
C          PRBLK(2)    TSTART, INITIAL TIME                           BLKDAT    22
C          PRBLK(3)    VTO, INITIAL VELOCITY MAGNITUDE                BLKDAT    23
C          PRBLK(4)    PHEADO, INITIAL PATH HEADING                   BLKDAT    24
C          PRBLK(5)    PPITCHO, INITIAL PATH PITCH                    BLKDAT    25
C          PRBLK(6)    ALFAO, INITIAL ALPHA ANGLE                     BLKDAT    26
C          PRBLK(7)    LATO, INITIAL LATITUDE                         BLKDAT    27
C          PRBLK(8)    LONO, INITIAL LONGITUDE                        BLKDAT    28
C          PRBLK(9)    ALTO, INITIAL ALTITUDE                         BLKDAT    29
C          PRBLK(10)   IPRNT, PRINT CONTROL PARAMETER                 BLKDAT    30
C          PRBLK(11)   IRITE, TAPE OUTPUT CONTROL PARAMETER           BLKDAT    31
C          PRBLK(12)   IPLOT, PLOT CONTROL PARAMETER                  BLKDAT    32
C          PRBLK(13)   ROLRATE, AIRCRAFT ROLL RATE                    BLKDAT    33
C                                                                     BLKDAT    34
C       COMMON BLOCK "SUPLE" CONTAINS SUPLEMENTARY VARIABLES          BLKDAT    35
C          SUPLE(1)    T, TIME                                        BLKDAT    36
C          SUPLE(2)    TF, FINAL TIME OF A SEGMENT                    BLKDAT    37
C          SUPLE(3)    TI, INITIAL TIME OF A SEGMENT                  BLKDAT    38
C          SUPLE(4)    TRNDONE, COMPLETED TURN INDICATOR              BLKDAT    39
C          SUPLE(5)    TDONE, TIME WHEN TURN COMPLETE                 BLKDAT    40
C          SUPLE(6)    ISEG, FLIGHT SEGMENT INDEX                     BLKDAT    41
C          SUPLE(7)    TOFF, TIME WHEN ROLL-UP STOPS                  BLKDAT    42
C          SUPLE(8)    TON, TIME WHEN ROLL-DOWN STARTS                BLKDAT    43
C          SUPLE(9)    RRCOEF, ROLL CONTROL COEFFICIENT               BLKDAT    44
C                                                                     BLKDAT    45
C       COMMON BLOCK "FIXFD" CONTAINS CONSTANTS                       BLKDAT    46
C          FIXFD(1)    N, DIMENSION OF STATE VECTOR                   BLKDAT    47
C          FIXFD(2)    RADPERD, RADIANS PER DEGREE                    BLKDAT    48
C          FIXFD(3)    TWOPI, 2*PI                                    BLKDAT    49
C          FIXFD(4)    PI                                             BLKDAT    50
C          FIXFD(5)    HALFPI, PI/2                                   BLKDAT    51
C          FIXED(6)    RE, EQUATORIAL EARTH RADIUS                    BLKDAT    52
C          FIXFD(7)    ESQ, EARTH ECCENTRICITY SQUARED                BLKDAT    53
C          FIXFD(8)    WEI, EARTH ANGULAR VELOCITY                    BLKDAT    54
C          FIXFD(9)    COEFFICIENT FOR LEVEL GRAVITY COMPONENTS       BLKDAT    55
C          FIXFD 10-15 COEFFICIENTS FOR RADIAL GRAVITY COMPONENT      BLKDAT    56
C                                                                     BLKDAT    57
C       THE NAV FRAME IS MECHANIZED AS LOCAL-LEVEL, ALPHA-WANDER.     BLKDAT    58
```

```
C        THE NAV FRAME IS ORIENTED AS FOLLOWS:                              BLKDAT     59
C           X - LIES IN A PLANE TANGENT TO THE REFERENCE ELLIPSOID: X IS   BLKDAT     60
C               ROTATED ALPHA DEGREES CCW FROM NORTH                        BLKDAT     61
C           Z - PERPENDICULAR TO THE REFERENCE ELLIPSOID AND POINTED UP    BLKDAT     62
C           Y - ORIENTED TO COMPLETE A RIGHT-HANDED, ORTHOGONAL TRIAD      BLKDAT     63
C                                                                           BLKDAT     64
C        THE PATH FRAME IS ORIENTED AS FOLLOWS:                             BLKDAT     65
C           X - LIES ALONG THE AIRCRAFT VELOCITY VECTOR                     BLKDAT     66
C           Y - POINTS OUT THE RIGHT WING                                   BLKDAT     67
C           Z - ORIENTED TO COMPLETE A RIGHT-HANDED, ORTHOGONAL TRIAD      BLKDAT     68
C                                                                           BLKDAT     69
                                                                            BLKDAT     70
      COMMON /FIXED/FIXED(15)                                               BLKDAT     71
      COMMON /SEGLNT/SEGLNT(50)  /RESTART/RESTART(50)                       BLKDAT     72
      COMMON /TURN/TURN(50)      /NPATH/NPATH(50)     /PACC/PACC(50)        BLKDAT     73
      COMMON /TACC/TACC(50)      /HEAD/HEAD(50)       /PITCH/PITCH(50)      BLKDAT     74
      COMMON /MODE/MODE(50)      /ERROR/ERROR(50)     /HMAX/HMAX(50)        BLKDAT     75
      COMMON /HMIN/HMIN(50)      /DTO/DTO(50)                               BLKDAT     76
                                                                            BLKDAT     77
      EQUIVALENCE (FIXED( 1),   N    )                                      BLKDAT     78
      EQUIVALENCE (FIXED( 2),RADPERD)                                       BLKDAT     79
      EQUIVALENCE (FIXED( 6),   RE   )                                      BLKDAT     80
      EQUIVALENCE (FIXED( 7),   ESQ  )                                      BLKDAT     81
      EQUIVALENCE (FIXED( 8),   WEI  )                                      BLKDAT     82
      EQUIVALENCE (FIXED( 9), GLHSC )                                       BLKDAT     83
      EQUIVALENCE (FIXED(10), GRCNT )                                       BLKDAT     84
      EQUIVALENCE (FIXED(11),  GRS2 )                                       BLKDAT     85
      EQUIVALENCE (FIXED(12),  GRS4 )                                       BLKDAT     86
      EQUIVALENCE (FIXED(13),  GRH  )                                       BLKDAT     87
      EQUIVALENCE (FIXED(14), GRHS2 )                                       BLKDAT     88
      EQUIVALENCE (FIXED(15),  GRH2 )                                       BLKDAT     89
                                                                            BLKDAT     90
      INTEGER RESTART,TURN                                                  BLKDAT     91
                                                                            BLKDAT     92
      DATA SEGLNT/50*0./, RESTART/50*0/, TURN/50*4/, NPATH/50*2/,           BLKDAT     93
     1     PACC/50*0./, TACC/50*0./, HEAD/50*0./, PITCH/50*0./,             BLKDAT     94
     2     MODE/50*1/, ERROR/50*1.E-06/, HMAX/50*1.E+4/, HMIN/50*1./,       BLKDAT     95
     3     DTO/50*1.E+08/                                                   BLKDAT     96
      DATA N/23/,                                                           BLKDAT     97
     1     RADPERD/1.7453292519943E-2/, RE/2.0925640E+7/,                   BLKDAT     98
     2     ESQ/6.6943177780E-3/,          WEI/7.292115147E-5/,              BLKDAT     99
     3     GLHSC/1.63E-8/,                                                  BLKDAT    100
     4     GRCNT/32.0877057/,           GRS2/0.16932081/,                   BLKDAT    101
     5     GRS4/7.528105E-4/,           GRH/3.6227E-8/,                     BLKDAT    102
     6     GRHS2/6.4089E-10/,           GRH2/6.8512E-15/                    BLKDAT    103
      END                                                                   BLKDAT    104
```

```
                 SUBROUTINE CHKSHC                                           CHKSHC       2
                                                                             CHKSHC       3
         C**   BEFORE EACH SINE-HEADING-CHANGE MANEUVER, CHKSHC CHECKS TO SEE IF    CHKSHC       4
         C**   THE MANEUVER CAN BE PERFORMED AS REQUESTED.  IF IT CAN'T, THE        CHKSHC       5
         C**   RUN IS TERMINATED AND A BRIEF EXPLANATORY MESSAGE IS PRINTED OUT.    CHKSHC       6
                                                                             CHKSHC       7
                 COMMON /FIXED/FIXED(15)                                     CHKSHC       8
                 COMMON /HEAD/HEAD(50)                                       CHKSHC       9
                 COMMON /HMIN/HMIN(50)                                       CHKSHC      10
                 COMMON /PITCH/PITCH(50)                                     CHKSHC      11
                 COMMON /PRBLK/PRBLK(13)                                     CHKSHC      12
                 COMMON /STATE/STATE(23)                                     CHKSHC      13
                 COMMON /SUPLE/SUPLE(9)                                      CHKSHC      14
                                                                             CHKSHC      15
                 EQUIVALENCE (FIXED(5),HALFPI)                               CHKSHC      16
                 EQUIVALENCE (PRBLK(13),ROLRATE)                             CHKSHC      17
                 EQUIVALENCE (STATE(4),VT)                                   CHKSHC      18
                 EQUIVALENCE (SUPLE(6),ISEG)                                 CHKSHC      19
                                                                             CHKSHC      20
                 TEST=VT*PITCH(ISEG)*PITCH(ISEG)*ABS(HEAD(ISEG))/16.1        CHKSHC      21
                 TEST=AMIN1(ROLRATE,TEST)                                    CHKSHC      22
                 TEST=TEST*HMIN(ISEG)                                        CHKSHC      23
                 IF (TEST.LT.HALFPI) RETURN                                  CHKSHC      24
                 WRITE (6,100)                                               CHKSHC      25
                 STOP                                                        CHKSHC      26
             100 FORMAT(//T2,*CHKSHC MESSAGE  -  THE PRODUCT OF COMPUTED ROLL RATE  CHKSHC      27
                1AND MINIMUM STEP SIZE EXCEEDS 90 DEGREES.*/T2,*BANK ANGLES IN EXCE CHKSHC      28
                2SS OF 90 DEGREES ARE NOT ALLOWED.  PROGRAM TERMINATED.*)            CHKSHC      29
                 END                                                         CHKSHC      30
```

```
      SUBROUTINE ETADOT(ETAXDOT,ETAYDOT,ETAZDOT)                        ETADOT   2
                                                                        ETADOT   3
C**   ETADOT COMPUTES ROLL RATE, PITCH RATE AND YAW RATE.               ETADOT   4
                                                                        ETADOT   5
      COMMON /STATE/STATE(23)                                           ETADOT   6
      EQUIVALENCE (STATE(8),SY)                                         ETADOT   7
      EQUIVALENCE (WPN(1),P)                                            ETADOT   8
      EQUIVALENCE (WPN(2),Q)                                            ETADOT   9
      EQUIVALENCE (WPN(3),R)                                            ETADOT  10
                                                                        ETADOT  11
      DIMENSION WPN(3)                                                  ETADOT  12
                                                                        ETADOT  13
      CY=COS(ETAY(DMY))                                                 ETADOT  14
      IF (CY.EQ.0.) GO TO 10                                            ETADOT  15
      ETZ=ETAZ(DMY)                                                     ETADOT  16
      SZ=SIN(ETZ)                                                       ETADOT  17
      CZ=COS(ETZ)                                                       ETADOT  18
      CALL OMEGAPN(WPN)                                                 ETADOT  19
      FACTOR=P*CZ-Q*SZ                                                  ETADOT  20
      ETAXDOT= FACTOR/CY                                                ETADOT  21
      ETAYDOT=-P*SZ-Q*CZ                                                ETADOT  22
      ETAZDOT=-R+(SY/CY)*FACTOR                                         ETADOT  23
      RETURN                                                            ETADOT  24
   10 ETAXDOT=ETAYDOT=ETAZDOT=0.                                        ETADOT  25
      WRITE (6,100)                                                     ETADOT  26
  100 FORMAT(T2,*ROLL AND YAW RATES ARE UNDEFINED WHEN PITCH IS 90 DEGRE ETADOT  27
     1ES. THUS ALL RATES HAVE BEEN TEMPORARILY ZEROED.*)                ETADOT  28
      RETURN                                                            ETADOT  29
      END                                                               ETADOT  30
```

```
FUNCTION ETAX       74/74    OPT=2                    FTN 4.5+414       06/11/76  13.22.47        PAGE    14

                REAL FUNCTION ETAX(DMY)                                        ETAX        2
                                                                               ETAX        3
          C**   ETAX COMPUTES THE PATH-TO-NAV ROLL ANGLE                       ETAX        4
                                                                               ETAX        5
                COMMON /STATE/STATE(23)                                        ETAX        6
                EQUIVALENCE (STATE(11),CPN32)                                  ETAX        7
                EQUIVALENCE (STATE(14),CPN33)                                  ETAX        8
                                                                               ETAX        9
                ETAX=ATAN2(-CPN32,-CPN33)                                      ETAX       10
                RETURN                                                         ETAX       11
                END                                                            ETAX       12
```

```
      REAL FUNCTION ETAY(DMY)                                  ETAY   2
                                                               ETAY   3
C**   ETAY COMPUTES THE PATH-TO-NAV PITCH ANGLE                ETAY   4
                                                               ETAY   5
      COMMON /STATE/STATE(23)                                  ETAY   6
      EQUIVALENCE (STATE(9),CPN31)                             ETAY   7
                                                               ETAY   8
      ETAY=ASIN(CPN31)                                         ETAY   9
      RETURN                                                   ETAY  10
      END                                                      ETAY  11
```

```
      REAL FUNCTION ETAZ(DMY)                                   ETAZ     2
                                                                ETAZ     3
C**   ETAZ COMPUTES THE PATH-TO-NAV YAW ANGLE                   ETAZ     4
                                                                ETAZ     5
      COMMON /STATE/STATE(23)                                   ETAZ     6
      EQUIVALENCE (STATE(6),CPN11)                              ETAZ     7
      EQUIVALENCE (STATE(7),CPN21)                              ETAZ     8
                                                                ETAZ     9
      ETAZ=ATAN2(-CPN21,CPN11)                                  ETAZ    10
      RETURN                                                    ETAZ    11
      END                                                       ETAZ    12
```

```
      SUBROUTINE F(N,TIME,Y,DY)

C**   THIS SUBROUTINE COMPUTES THE DERIVATIVES THAT ARE
C**   NUMERICALLY INTEGRATED IN SUBROUTINE KUTMER.

      COMMON /PACC/PACC(50)
      COMMON /STATE/STATE(23)
      COMMON /SUPLE/SUPLE(9)

      EQUIVALENCE (STATE(3),VZ)
      EQUIVALENCE (STATE(6),CPN(1,1) )
      EQUIVALENCE (STATE(15),CEN(1,1) )
      EQUIVALENCE (SUPLE(1),T)
      EQUIVALENCE (SUPLE(6),ISEG)

      DIMENSION Y(N),DY(N)
      DIMENSION WPN(3),SWPN(3,3),CPN(3,3),CPNDOT(3,3)
      DIMENSION RHO(3),SRHO(3,3),CEN(3,3),CENDOT(3,3)

C
C**   ADVANCE TIME AND UPDATE STATE VECTOR TO AGREE WITH PROGRESS
C**   IN KUTMER
C
      T=TIME
      DO 10 I=1,N
   10 STATE(I)=Y(I)
C
C**   DERIVATIVE COMPUTATIONS
C
      CALL OMEGAPN(WPN)
      CALL RHONE(RHO)
      DO 20 I=1,3
   20 RHO(I)=-RHO(I)
      CALL SKEW(WPN,SWPN)
      CALL SKEW(RHO,SRHO)
      CALL AXB(SWPN,CPN,CPNDOT,3,3,3)
      CALL AXB(SRHO,CEN,CENDOT,3,3,3)
      CALL VDOT(DVX,DVY,DVZ)
C
C**   FILL DY FOR RETURN TO KUTMER
C
      DY(1)=DVX
      DY(2)=DVY
      DY(3)=DVZ
      DY(4)=PACC(ISEG)
      DY(5)=VZ
      DY(6)=CPNDOT(1,1)
      DY(7)=CPNDOT(2,1)
      DY(8)=CPNDOT(3,1)
      DY(9)=CPNDOT(1,2)
      DY(10)=CPNDOT(2,2)
      DY(11)=CPNDOT(3,2)
      DY(12)=CPNDOT(1,3)
      DY(13)=CPNDOT(2,3)
      DY(14)=CPNDOT(3,3)
      DY(15)=CENDOT(1,1)
      DY(16)=CENDOT(2,1)
```

```
      DY(17)=CENDOT(3,1)
      DY(18)=CENDOT(1,2)
      DY(19)=CENDOT(2,2)
      DY(20)=CENDOT(3,2)
      DY(21)=CENDOT(1,3)
      DY(22)=CENDOT(2,3)
      DY(23)=CENDOT(3,3)
      RETURN
      END
```

```
      SUBROUTINE GRAVITY(GX,GY,GZ)                                       GRAVITY    2
                                                                         GRAVITY    3
C**   GRAVITY COMPUTES THE THREE COMPONENTS OF THE EARTH'S PLUMB-BOB     GRAVITY    4
C**   GRAVITY VECTOR, A VECTOR THAT CONSISTS OF BOTH MASS ATTRACTION     GRAVITY    5
C**   AND CENTRIPETAL COMPONENTS.                                        GRAVITY    6
                                                                         GRAVITY    7
      COMMON /FIXED/FIXED(15)                                            GRAVITY    8
      COMMON /STATE/STATE(23)                                            GRAVITY    9
                                                                         GRAVITY   10
      EQUIVALENCE (FIXED( 9),GLHSC)                                      GRAVITY   11
      EQUIVALENCE (FIXED(10),GRCNT)                                      GRAVITY   12
      EQUIVALENCE (FIXED(11),GRS2)                                       GRAVITY   13
      EQUIVALENCE (FIXED(12),GRS4)                                       GRAVITY   14
      EQUIVALENCE (FIXED(13),GRH)                                        GRAVITY   15
      EQUIVALENCE (FIXED(14),GRHS2)                                      GRAVITY   16
      EQUIVALENCE (FIXED(15),GRH2)                                       GRAVITY   17
      EQUIVALENCE (STATE( 5),ALT)                                        GRAVITY   18
      EQUIVALENCE (STATE(15),CEN11)                                      GRAVITY   19
      EQUIVALENCE (STATE(16),CEN21)                                      GRAVITY   20
      EQUIVALENCE (STATE(17),CEN31)                                      GRAVITY   21
                                                                         GRAVITY   22
      S2PHI=CEN31*CEN31                                                  GRAVITY   23
      COEF=-GLHSC*ALT*CEN31                                              GRAVITY   24
      GX=COEF*CEN11                                                      GRAVITY   25
      GY=COEF*CEN21                                                      10JUN76    1
      GZ=-(GRCNT+GRS2*S2PHI+GRS4*S2PHI*S2PHI) *                          GRAVITY   27
     1    (1.0-(GRH-GRHS2*S2PHI)*ALT+GRH2*ALT*ALT)                       GRAVITY   28
      RETURN                                                             GRAVITY   29
      END                                                                GRAVITY   30
```

```
      REAL FUNCTION HCHOP(H,T,TEVENT)                                 HCHOP     2
                                                                      HCHOP     3
C**   HCHOP BEGINS BY COMPUTING THE TIME INTERVAL FROM T (PRESENT     HCHOP     4
C**   TIME) TO TEVENT (A FUTURE TIME WHEN SOME EVENT MUST OCCUR).     HCHOP     5
C**   IF THE PLANNED INTEGRATION STEP, H, WILL CARRY T BEYOND         HCHOP     6
C**   TEVENT, A NEW STEP, HCHOP, IS COMPUTED SO THAT                  HCHOP     7
C**                T + HCHOP = TEVENT                                 HCHOP     8
C**   TO PREVENT LOSS OF SIGNIFICANCE IN KUTMER WHERE T ADDED TO H/3  HCHOP     9
C**   MUST BE GREATER THAN T, HCHOP MUST BE KEPT ABOVE A WORKING      HCHOP    10
C**   MINIMUM. FOR THE 48 BIT MANTISSA OF THE 60 BIT CDC WORD,        HCHOP    11
C**   THE ABSOLUTE MINIMUM WOULD BE T*(2**-48)/3=T*(1.2*10**-15).     HCHOP    12
C**   TO BE CONSERVATIVE THE WORKING MINIMUM WAS SET TO T*(10**-13).  HCHOP    13
                                                                      HCHOP    14
      HCHOP=H                                                         HCHOP    15
      IF (T.GT.TEVENT) RETURN                                         HCHOP    16
      HNOM=TEVENT-T                                                   HCHOP    17
      HMIN=ABS(T)*(1.E-13)                                            HCHOP    18
      HNOM=AMAX1(HMIN,HNOM)                                           HCHOP    19
      HCHOP=AMIN1(H,HNOM)                                             HCHOP    20
      RETURN                                                          HCHOP    21
      END                                                             HCHOP    22
```

```
      SUBROUTINE HEADER                                                 HEADER   2
                                                                        HEADER   3
C**   HEADER PRINTS A DESCRIPTION OF EACH SEGMENT                       HEADER   4
C**   AT TI, THE INITIAL TIME OF THE SEGMENT.                           HEADER   5
                                                                        HEADER   6
      COMMON /MODE/MODE(50)                                             HEADER   7
      COMMON /NPATH/NPATH(50)                                           HEADER   8
      COMMON /PRBLK/PRBLK(13)                                           HEADER   9
      COMMON /SUPLE/SUPLE(9)                                            HEADER  10
      COMMON /TURN/TURN(50)                                             HEADER  11
                                                                        HEADER  12
      EQUIVALENCE (PRBLK(1),LLMECH)                                     HEADER  13
      EQUIVALENCE (SUPLE(1),T)                                          HEADER  14
      EQUIVALENCE (SUPLE(2),TF)                                         HEADER  15
      EQUIVALENCE (SUPLE(6),ISEG)                                       HEADER  16
                                                                        HEADER  17
      INTEGER TURN                                                      HEADER  18
      DIMENSION AA(8),BB(4),CC(2),DD(8)                                 HEADER  19
                                                                        HEADER  20
      DATA (AA(I), I=1,8) /10H VERTICAL ,5HTURN.,10H HORIZONTA,7HL TURN. HEADER  21
     1,10H SINE HEAD,10HING CHANGE,10H STRAIGHT ,7HFLIGHT.  /,          HEADER  22
     2 (BB(J),J=1,4) /10H GREAT CIR,4HCLE.,10H RHUMB LIN,2HE. /,        HEADER  23
     3 (CC(K),K=1,2) /7H FIXED.,10H VARIABLE. /,                        HEADER  24
     4 (DD(L),L=1,8) /10H AZIMUTH W,6HANDER.,10H CONSTANT ,6HALPHA.,    HEADER  25
     5 10H UNIPOLAR.,1H ,10H FREE AZIM,4HUTH./                          HEADER  26
                                                                        HEADER  27
      IA=2*TURN(ISEG)-1                                                 HEADER  28
      IB=2*NPATH(ISEG)-1                                                HEADER  29
      IC=MODE(ISEG)+1                                                   HEADER  30
      ID=2*LLMECH-1                                                     HEADER  31
      WRITE (6,100) ISEG,T,TF,AA(IA),AA(IA+1),BB(IB),BB(IB+1),          HEADER  32
     1CC(IC),DD(ID),DD(ID+1)                                            HEADER  33
  100 FORMAT (1H1,T5,*BEGIN SEGMENT NUMBER*,I3/                         HEADER  34
     1          T5,*INITIAL TIME *,T18,F12.5/                           HEADER  35
     2          T5,*FINAL TIME *,T18,F12.5/                             HEADER  36
     3          T5,*THIS FLIGHT SEGMENT IS A*,2A10/                     HEADER  37
     4          T5,*THE NOMINAL FLIGHT PATH OVER THE EARTH IS A*,2A10/  HEADER  38
     5          T5,*THE INTEGRATION STEP SIZE IS*,A10/                  HEADER  39
     6          T5,*THE LOCAL LEVEL MECHANIZATION IS*,2A10)             HEADER  40
      RETURN                                                            HEADER  41
      END                                                               HEADER  42
```

```
      REAL FUNCTION HLIMIT(T,TF,H,HMN)

C**   HLIMIT BEGINS BY RAISING THE STEP SIZE TO AN OPERATING MINIMUM.
C**   HLIMIT ADJUSTS THE STEP SIZE SO THAT THE PROGRAM WILL NOT STEP
C**   PAST THE END OF A FLIGHT SEGMENT OR PAST A REQUIRED OUTPUT TIME.

      HLIMIT=AMAX1(H,HMN)
      HLIMIT=HCHOP(HLIMIT,T,TF)
      HLIMIT=HCHOP(HLIMIT,T,TOUT(DMY))
      RETURN
      END
```

```
      SUBROUTINE HLIM2(H,RRCOEF)

C**   HLIM2 ADJUSTS THE STEP SIZE IN A HORIZONTAL TURN SO THAT THE
C**   PROGRAM WILL PAUSE AT POINTS WHERE THE AIRCRAFT IS FINISHING OR
C**   BEGINNING A ROLL MANEUVER, I.E. AT TOFF, TDONE AND TON. HLIM2 ALSO
C**   SETS THE ROLL CONTROL COEFFICIENT FOR MAKING ROLRATE PLUS, ZERO
C**   OR MINUS IN SUBROUTINE ROLDOTC.

      COMMON /SUPLE/SUPLE(9)
      EQUIVALENCE (SUPLE(1),T)
      EQUIVALENCE (SUPLE(2),TF)
      EQUIVALENCE (SUPLE(5),TDONE)
      EQUIVALENCE (SUPLE(7),TOFF)
      EQUIVALENCE (SUPLE(8),TON)

C
C             TRANSFER TO PROPER SUBSEGMENT
      IF (T.LT.TOFF) GO TO 10
      IF (T.GE.TOFF  .AND. T.LT.TON) GO TO 20
      IF (T.GE.TON   .AND. T.LT.TDONE) GO TO 40
      IF (T.GE.TDONE .AND. T.LT.TF) GO TO 50
C
C             SET RRCOEF AND LIMIT H IF NECESSARY
   10 RRCOEF=+1.
      H=HCHOP(H,T,TOFF)
      RETURN
   20 RRCOEF=0.
      H=HCHOP(H,T,TON)
      RETURN
   40 RRCOEF=-1.
      H=HCHOP(H,T,TDONE)
      RETURN
   50 RRCOEF=0.
      RETURN
      END
```

```
      SUBROUTINE HLIM3(H,RRCOEF)                                   HLIM3       2
                                                                   HLIM3       3
C**  HLIM3 ADJUSTS THE STEP SIZE IN A SINE HEADING MANEUVER SO     HLIM3       4
C**  THAT THE PROGRAM WILL PAUSE EACH HALF-PERIOD.  HLIM3 ALSO     HLIM3       5
C**  SETS THE ROLL CONTROL COEFFICIENT FOR MAKING ROLL RATE PLUS   HLIM3       6
C**  OR MINUS IN SUBROUTINE ROLDOTC.                               HLIM3       7
                                                                   HLIM3       8
      COMMON /FIXED/FIXED(15)                                      HLIM3       9
      COMMON /PITCH/PITCH(50)                                      HLIM3      10
      COMMON /SUPLE/SUPLE(9)                                       HLIM3      11
                                                                   HLIM3      12
      EQUIVALENCE (FIXED(4),PI)                                    HLIM3      13
      EQUIVALENCE (SUPLE(1),T)                                     HLIM3      14
      EQUIVALENCE (SUPLE(3),TI)                                    HLIM3      15
      EQUIVALENCE (SUPLE(6),ISEG)                                  HLIM3      16
                                                                   HLIM3      17
C                                                                  HLIM3      18
C           LIMIT H SO INTEGRATOR DOES NOT ATTEMPT TO              HLIM3      19
C           STEP PAST A HALF-PERIOD DEMARCATION POINT              HLIM3      20
      DT=T-TI                                                      HLIM3      21
      HP=PI/ABS(PITCH(ISEG))                                       HLIM3      22
      THP=TI+HP*(1.+AINT(DT/HP))                                   HLIM3      23
      H=HCHOP(H,T,THP)                                             HLIM3      24
C                                                                  HLIM3      25
C           SET ROLL CONTROL COEFFICIENT                           HLIM3      26
      M1=INT((DT+H/2.)/HP)                                         HLIM3      27
      MOE=MOD(M1,2)                                                HLIM3      28
      IF (MOE.EQ.0) RRCOEF=+1.                                     HLIM3      29
      IF (MOE.EQ.1) RRCOEF=-1.                                     HLIM3      30
      RETURN                                                       HLIM3      31
      END                                                          HLIM3      32
```

```
            SUBROUTINE KMPERF                                          KMPERF     2
                                                                       KMPERF     3
      C**   KMPERF PRINTS A SHORT SUMMARY OF THE                       KMPERF     4
      C**   NUMERICAL INTEGRATOR'S PERFORMANCE.                        KMPERF     5
                                                                       KMPERF     6
            COMMON /IKUT/IK1,IK2                                       KMPERF     7
                                                                       KMPERF     8
            IK3=5*IK2                                                  KMPERF     9
            WRITE (6,100) IK1,IK3                                      KMPERF    10
        100 FORMAT(///T5,*THIS FLIGHT REQUIRED*,I10,5X,   *PASSES THROUGH THE  KMPERF    11
           1NUMERICAL INTEGRATOR, KUTMER.*/T5,*KUTMER IN TURN MADE*,I11,5X,    KMPERF    12
           2*CALLS TO SUBROUTINE F, THE DERIVATIVE SUBPROGRAM.*)       KMPERF    13
            RETURN                                                     KMPERF    14
            END                                                        KMPERF    15
```

```
      SUBROUTINE KUTMER(N,TO,X,H,F,MODE,ERROR,HMAX,HMIN)                KUTMER     2
C                                                                       KUTMER     3
C     ................................................................. KUTMER     4
C                                                                       KUTMER     5
C        SUBROUTINE KUTMER                                              KUTMER     6
C                                                                       KUTMER     7
C        PURPOSE                                                        KUTMER     8
C           TO INTEGRATE A (POSSIBLY NONLINEAR) SET OF FIRST-ORDER      KUTMER     9
C           DIFFERENTIAL EQUATIONS FROM "TO" TO "TO+H"                  KUTMER    10
C                                                                       KUTMER    11
C        REFERENCE                                                      KUTMER    12
C           "AN EFFICIENCY STUDY OF SEVERAL TECHNIQUES FOR THE          KUTMER    13
C           NUMERICAL INTEGRATION OF THE EQUATIONS OF MOTION FOR        KUTMER    14
C           MISSILES AND SHELL", BY HAROLD J. BREAUX, FEBRUARY,1967,    KUTMER    15
C           AD 812362.                                                  KUTMER    16
C                                                                       KUTMER    17
C        USAGE                                                          KUTMER    18
C           CALL KUTMER(N,TO,X,H,F,MODE.ERROR,HMAX,HMIN)                KUTMER    19
C                                                                       KUTMER    20
C        DESCRIPTION OF PARAMETERS                                      KUTMER    21
C           N  - NUMBER OF DIFFERENTIAL EQUATIONS (MAX = 25)            KUTMER    22
C           TO - INITIAL TIME (DESTROYED). CONTAINS T-FINAL ON RETURN.  KUTMER    23
C           X  - INITIAL STATE VECTOR (DESTROYED). CONTAINS FINAL STATE KUTMER    24
C                ON RETURN.                                             KUTMER    25
C           H  - STEP SIZE. IF VARIABLE STEP SIZE OPTION IS USED,       KUTMER    26
C                H CONTAINS ADJUSTED STEP SIZE ON RETURN.               KUTMER    27
C              - EXTERNAL SUBROUTINE F(N,T,X,DX) CONTAINING N           KUTMER    28
C                DIFFERENTIAL EQUATIONS IN THE FORM DX=F(T,X). THIS     KUTMER    29
C                SUBROUTINE NAME MUST BE DECLARED AN EXTERNAL IN THE    KUTMER    30
C                PROGRAM THAT CALLS KUTMER.                             KUTMER    31
C           MODE  - IF MODE=1 THE STEP SIZE IS "VARIABLE", I.E. H IS    KUTMER    32
C                   ADJUSTED AUTOMATICALLY TO MAINTAIN THE INTEGRATION  KUTMER    33
C                   ERROR BELOW ITS ALLOWED VALUE.                      KUTMER    34
C                 - IF MODE NOT EQUAL 1, STEP SIZE IS "FIXED".          KUTMER    35
C           ERROR - ALLOWED INTEGRATION ERROR PER STEP WHEN MODE=1.     KUTMER    36
C           HMAX  - MAXIMUM STEP SIZE                                   KUTMER    37
C           HMIN  - MINIMUM STEP SIZE                                   KUTMER    38
C                                                                       KUTMER    39
C        EQUATIONS                                                      KUTMER    40
C           YO=X(TO)                                                    KUTMER    41
C           Y1=YO+(H/3)F(TO,YO)                                         KUTMER    42
C           Y2=YO+(H/6)F(TO,YO)+(1H/6)F(TO+H/3,Y1)                      KUTMER    43
C           Y3=YO+(H/8)F(TO,YO)+(3H/8)F(TO+H/3,Y2)                      KUTMER    44
C           Y4=YO+(H/2)F(TO,YO)-(3H/2)F(TO+H/3,Y2)+(2H)F(TO+H/2,Y3)     KUTMER    45
C           Y5=YO+(H/6)F(TO,YO)+(2H/3)F(TO+H/2,Y3)+(H/6)F(TO+H,Y4)      KUTMER    46
C             =X(TO+H)                                                  KUTMER    47
C                                                                       KUTMER    48
C        REMARKS                                                        KUTMER    49
C           SUBROUTINE F CAN DESTROY X WITHOUT AFFECTING KUTMER.        KUTMER    50
C           BOTH FOURTH ORDER AND FIFTH ORDER INTEGRATIONS ARE          KUTMER    51
C           PERFORMED. IF STEP SIZE IS FIXED, THE FIFTH ORDER ANSWER IS KUTMER    52
C           RETURNED IMMEDIATELY. IF STEP SIZE IS VARIABLE, THE FIFTH   KUTMER    53
C           ORDER ANSWER IS SUBTRACTED FROM THE FOURTH ORDER ANSWER     KUTMER    54
C           AND THE DIFFERENCE IS CHECKED AGAINST THE ERROR CRITERION.  KUTMER    55
C           IF THE ERROR IS IN BOUNDS, THE STEP SIZE IS INCREASED PRIOR KUTMER    56
C           TO RETURNING THE FIFTH ORDER ANSWER. IF THE ERROR IS OUT OF KUTMER    57
C           BOUNDS, THE STEP SIZE IS REDUCED, THE INTEGRATION IS        KUTMER    58
```

```
C           REDONE AND THE ERROR CHECKING PROCESS IS REPEATED.  THIS
C           SEQUENCE (LOWER H, INTEGRATE, CHECK ERROR) CONTINUES UNTIL
C           THE ERROR CRITERION IS SATISFIED OR UNTIL HMN IS REACHED.
C           IN THE LATTER CASE, A WARNING MESSAGE IS PRINTED BEFORE
C           RETURNING THE BEST AVAILABLE ANSWER.  H, HMAX, HMIN, AND
C           ERROR MUST ALL BE POSITIVE.
C
C        SUBROUTINES REQUIRED
C           SUBROUTINE F(N,T,X,DX) MUST BE FURNISHED BY THE USER.
C
C     ..................................................................
C
      COMMON /IKUT/IK1,IK2
      DIMENSION X(N),YO(25),Y(25,5)
      DATA IK1,IK2/2*0/

C    SAVE X(TO) IN YO AND "TO" IN TI TO INSULATE THEM AGAINST
C    DESTRUCTION BY SUBROUTINE F.  ALSO FIND HMN AND INCREMENT A
C    COUNTER TO KEEP TRACK OF CALLS TO KUTMER.
         DO 10 I=1,N
 10      YO(I)=X(I)
         TI=TO
         HMN=AMIN1(HMIN,H)
         IK1=IK1+1
C
C  ********** SOLVE FOR Y1 THROUGH Y5 **********
C
 15      T=TI
         IK2=IK2+1
C    OBTAIN F(TO,YO), COMPUTE Y1 AND ADVANCE TIME
         CALL F(N,T,X(1),Y(1,1))
         DO 20 I=1,N
 20      Y(I,4)=YO(I)+H*Y(I,1)/3.
         T=TI+H/3.
C    OBTAIN F(TO+H/3,Y1) AND COMPUTE Y2
         CALL F(N,T,Y(1,4),Y(1,2))
         DO 25 I=1,N
 25      Y(I,4)=YO(I)+H*Y(I,1)/6.+H*Y(I,2)/6.
C    OBTAIN F(TO+H/3,Y2), COMPUTE Y3 AND ADVANCE TIME
         CALL F(N,T,Y(1,4),Y(1,2))
         DO 30 I=1,N
 30      Y(I,4)=YO(I)+H*Y(I,1)/8.+.375*H*Y(I,2)
         T=TI+H/2.
C    OBTAIN F(TO+H/2,Y3), COMPUTE Y4 AND ADVANCE TIME
         CALL F(N,T,Y(1,4),Y(1,3))
         DO 35 I=1,N
 35      Y(I,4)=YO(I)+H*Y(I,1)/2.-1.5*H*Y(I,2)+2.*H*Y(I,3)
         T=TI+H
C    OBTAIN F(TO+H,Y4) AND COMPUTE Y5
         CALL F(N,T,Y(1,4),Y(1,2))
         DO 40 I=1,N
 40      Y(I,5)=YO(I)+H*Y(I,1)/6.+2.*H*Y(I,3)/3.+H*Y(I,2)/6.
         IF (MODE.NE.1) GO TO 75
C
C  ********** VARIABLE STEP-SIZE COMPUTATIONS **********
C
C    COMPUTE P, THE LARGEST DIFFERENCE BETWEEN
```

```
      C    CORRESPONDING ELEMENTS OF Y4 AND Y5.                         KUTMER   116
              P=0.                                                      KUTMER   117
              DO 45 I=1,N                                               KUTMER   118
       45     P=AMAX1(P,ABS(Y(I,4)-Y(I,5)))                             KUTMER   119
      C    TRANSFER TO STEP-SIZE-ADJUSTMENT COMPUTATION                 KUTMER   120
      C    ACCORDING TO RELATIVE MAGNITUDES OF P AND ERROR              KUTMER   121
              IF (P.EQ.0.) GO TO 50                                     KUTMER   122
              IF (P.LE.ERROR) GO TO 55                                  KUTMER   123
              IF (P.GT.ERROR) GO TO 60                                  KUTMER   124
      C    DOUBLE STEP SIZE                                             KUTMER   125
       50     IF (H.EQ.HMAX) GO TO 75                                   KUTMER   126
              H=2.*H                                                    KUTMER   127
              IF (H.GT.HMAX) H=HMAX                                     KUTMER   128
              GO TO 75                                                  KUTMER   129
      C    INCREASE STEP SIZE USING ALGORITHM                           KUTMER   130
       55     IF (H.EQ.HMAX) GO TO 75                                   KUTMER   131
              H=H*(ERROR/P)**0.2                                        KUTMER   132
              IF (H.GT.HMAX) H=HMAX                                     KUTMER   133
              GO TO 75                                                  KUTMER   134
      C    REDUCE H AND PREPARE TO REPEAT THE ENTIRE NUMERICAL INTEGRATION KUTMER 135
       60     IF (H.EQ.HMN) GO TO 70                                    KUTMER   136
              H=H*(0.1*ERROR/P)**0.2                                    KUTMER   137
              IF (H.LT.HMN) H=HMN                                       KUTMER   138
              DO 65 I=1,N                                               KUTMER   139
       65     X(I)=YO(I)                                                KUTMER   140
              GO TO 15                                                  KUTMER   141
       70     WRITE (6,100)                                             KUTMER   142
      C                                                                 KUTMER   143
      C  ********* FILL X VECTOR AND ADVANCE TIME FOR RETURN *********   KUTMER   144
      C                                                                 KUTMER   145
       75     DO 80 I=1,N                                               KUTMER   146
       80     X(I)=Y(I,5)                                               KUTMER   147
              TO=T                                                      KUTMER   148
              RETURN                                                    KUTMER   149
      100     FORMAT(/T5,*THE INTEGRATION ERROR EXCEEDS ITS ALLOWED VALUE*) KUTMER 150
              END                                                       KUTMER   151
```

```
      REAL FUNCTION LAMDA(DMY)                               LAMDA       2
C**   LAMDA COMPUTES LONGITUDE.                              LAMDA       3
                                                             LAMDA       4
      COMMON /STATE/STATE(23)                                LAMDA       5
      EQUIVALENCE (STATE(20),CEN32)                          LAMDA       6
      EQUIVALENCE (STATE(23),CEN33)                          LAMDA       7
                                                             LAMDA       8
      LAMDA=ATAN2(-CEN32,CEN33)                              LAMDA       9
      RETURN                                                 LAMDA      10
      END                                                    LAMDA      11
                                                             LAMDA      12
```

```
      REAL FUNCTION LAMDOT(DMY)                                    LAMDOT     2
C**   LAMDOT COMPUTES THE ANGULAR RATE-OF-CHANGE OF LONGITUDE.       LAMDOT     3
                                                                   LAMDOT     4
      COMMON /STATE/STATE(23)                                      LAMDOT     5
      EQUIVALENCE (STATE(5),ALT)                                   LAMDOT     6
                                                                   LAMDOT     7
      LAMDOT=VEAST(DMY)/((RP(DMY)+ALT)*COS(PHI(DMY)))              LAMDOT     8
      RETURN                                                       LAMDOT     9
      END                                                          LAMDOT    10
                                                                   LAMDOT    11
```

```
SUBROUTINE MAXMIN     74/74    OPT=2                 FTN 4.5+414        06/11/76  13.22.47        PAGE    31

                SUBROUTINE MAXMIN(XARRAY,YARRAY,XMAX,XMIN,YMAX,YMIN)           MAXMIN      2
         CC                                                                    MAXMIN      3
         CC      DETERMINES THE MAXIMUM AND MINIMUM FOR XARRAY AND             MAXMIN      4
         CC      YARRAY FOR PLOTTING PURPOSES.                                 MAXMIN      5
         CC                                                                    MAXMIN      6
                COMMON /NPLOT/NPLTPTS,NPLTSEG(50)                              MAXMIN      7
                                                                               MAXMIN      8
                DIMENSION XARRAY(1001),YARRAY(1001)                            MAXMIN      9
                                                                               MAXMIN     10
                XMAX = XARRAY(1)                                               MAXMIN     11
                YMAX = YARRAY(1)                                               MAXMIN     12
                DO 10 I=2,NPLTPTS                                              MAXMIN     13
                IF (XARRAY(I).GT.XMAX)XMAX=XARRAY(I)                           MAXMIN     14
                IF (YARRAY(I).GT.YMAX)YMAX=YARRAY(I)                           MAXMIN     15
         10     CONTINUE                                                       MAXMIN     16
                WRITE(6,1000) XMAX,YMAX                                        MAXMIN     17
                XMIN = XARRAY(1)                                               MAXMIN     18
                YMIN = YARRAY(1)                                               MAXMIN     19
                DO 20 I=2,NPLTPTS                                              MAXMIN     20
                IF (XARRAY(I).LT.XMIN)XMIN=XARRAY(I)                           MAXMIN     21
                IF (YARRAY(I).LT.YMIN)YMIN=YARRAY(I)                           MAXMIN     22
         20     CONTINUE                                                       MAXMIN     23
                WRITE(6,1100) XMIN,YMIN                                        MAXMIN     24
         1000   FORMAT(2X,"XMAX = ",1PE13.6,2X,"YMAX = ",1PE13.6/)             MAXMIN     25
         1100   FORMAT(2X,"XMIN = ",1PE13.6,2X,"YMIN = ",1PE13.6/)             MAXMIN     26
                END                                                            MAXMIN     27
```

```
SUBROUTINE NEWUNIT     74/74    OPT=2                     FTN 4.5+414        06/11/76  13.22.47        PAGE     32

          SUBROUTINE NEWUNIT                                                   NEWUNIT     2
                                                                               NEWUNIT     3
C**   NEWUNIT CONVERTS INPUT DATA IN DEGREES AND G'S TO DATA                   NEWUNIT     4
C**   IN RADIANS AND FEET/SEC./SEC. RESPECTIVELY.                              NEWUNIT     5
                                                                               NEWUNIT     6
          COMMON /FIXED/FIXED(15)                                              NEWUNIT     7
          COMMON /HEAD/HEAD(50)                                                NEWUNIT     8
          COMMON /PACC/PACC(50)                                                NEWUNIT     9
          COMMON /PITCH/PITCH(50)                                              NEWUNIT    10
          COMMON /PRBLK/PRBLK(13)                                              NEWUNIT    11
          COMMON /TACC/TACC(50)                                                NEWUNIT    12
                                                                               NEWUNIT    13
          EQUIVALENCE (FIXED(2),RADPERD)                                       NEWUNIT    14
          EQUIVALENCE (PRBLK(4),PHEADO)                                        NEWUNIT    15
          EQUIVALENCE (PRBLK(5),PPITCHO)                                       NEWUNIT    16
          EQUIVALENCE (PRBLK(6),ALFAO)                                         NEWUNIT    17
          EQUIVALENCE (PRBLK(7),LATO)                                          NEWUNIT    18
          EQUIVALENCE (PRBLK(8),LONO)                                          NEWUNIT    19
          EQUIVALENCE (PRBLK(13),ROLRATE)                                      NEWUNIT    20
                                                                               NEWUNIT    21
          REAL LATO,LONO                                                       NEWUNIT    22
                                                                               NEWUNIT    23
          DO 10 I=4,8                                                          NEWUNIT    24
   10     PRBLK(I)=PRBLK(I)*RADPERD                                            NEWUNIT    25
          ROLRATE=ROLRATE*RADPERD                                              NEWUNIT    26
          DO 20 I=1,50                                                         NEWUNIT    27
          PACC(I)=PACC(I)*32.2                                                 NEWUNIT    28
   20     TACC(I)=TACC(I)*32.2                                                 NEWUNIT    29
          DO 30 I=1,50                                                         NEWUNIT    30
          HEAD(I)=HEAD(I)*RADPERD                                              NEWUNIT    31
   30     PITCH(I)=PITCH(I)*RADPERD                                            NEWUNIT    32
          RETURN                                                               NEWUNIT    33
          END                                                                  NEWUNIT    34
```

```
      SUBROUTINE OMEGAPN(WPN)                                           OMEGAPN     2
                                                                        OMEGAPN     3
C**   OMEGAPN SPECIFIES THE NAV FRAME COMPONENTS OF WPN (THE ANGULAR    OMEGAPN     4
C**   VELOCITY OF THE PATH FRAME WITH RESPECT TO THE NAV FRAME)         OMEGAPN     5
C**   THAT ARE REQUIRED TO EXECUTE THE MANEUVER.                        OMEGAPN     6
                                                                        OMEGAPN     7
      COMMON /PITCH/PITCH(50)                                           OMEGAPN     8
      COMMON /PRBLK/PRBLK(13)                                           OMEGAPN     9
      COMMON /STATE/STATE(23)                                           OMEGAPN    10
      COMMON /SUPLE/SUPLE(9)                                            OMEGAPN    11
      COMMON /TACC/TACC(50)                                             OMEGAPN    12
      COMMON /TURN/TURN(50)                                             OMEGAPN    13
                                                                        OMEGAPN    14
      EQUIVALENCE (PRBLK(1),LLMECH)                                     OMEGAPN    15
      EQUIVALENCE (STATE(4),VT)                                         OMEGAPN    16
      EQUIVALENCE (STATE(6),CPN11)                                      OMEGAPN    17
      EQUIVALENCE (STATE(7),CPN21)                                      OMEGAPN    18
      EQUIVALENCE (STATE(8),CPN31)                                      OMEGAPN    19
      EQUIVALENCE (STATE(9),CPN12)                                      OMEGAPN    20
      EQUIVALENCE (STATE(10),CPN22)                                     OMEGAPN    21
      EQUIVALENCE (STATE(11),CPN32)                                     OMEGAPN    22
      EQUIVALENCE (SUPLE(4),TRNDONE)                                    OMEGAPN    23
      EQUIVALENCE (SUPLE(6),ISEG)                                       OMEGAPN    24
                                                                        OMEGAPN    25
      INTEGER TURN                                                      OMEGAPN    26
      DIMENSION WPN(3)                                                  OMEGAPN    27
                                                                        OMEGAPN    28
C                                                                       OMEGAPN    29
C             YAW INDUCED PORTION                                       OMEGAPN    30
      WPN(1)=0.                                                         OMEGAPN    31
      WPN(2)=0.                                                         OMEGAPN    32
      WPN(3)=-ALFADOT(LLMECH)-PSIDOT(DMY)                               OMEGAPN    33
      IF (TURN(ISEG).EQ.4) RETURN                                       OMEGAPN    34
      IF (TURN(ISEG).EQ.1) GO TO 10                                     OMEGAPN    35
C                                                                       OMEGAPN    36
C             ROLL INDUCED PORTION                                      OMEGAPN    37
      DROLL=ROLDOTC(TURN(ISEG))                                         OMEGAPN    38
      WPN(1)=CPN11*DROLL                                                OMEGAPN    39
      WPN(2)=CPN21*DROLL                                                OMEGAPN    40
      WPN(3)=CPN31*DROLL+WPN(3)                                         OMEGAPN    41
      RETURN                                                            OMEGAPN    42
C                                                                       OMEGAPN    43
C             PITCH INDUCED PORTION                                     OMEGAPN    44
   10 AN=(1.-TRNDONE)*SIGN(1.,PITCH(ISEG))*TACC(ISEG)                   OMEGAPN    45
      DPITCH=AN/VT                                                      OMEGAPN    46
      WPN(1)=CPN12*DPITCH                                               OMEGAPN    47
      WPN(2)=CPN22*DPITCH                                               OMEGAPN    48
      WPN(3)=CPN32*DPITCH+WPN(3)                                        OMEGAPN    49
      RETURN                                                            OMEGAPN    50
      END                                                               OMEGAPN    51
```

```
            REAL FUNCTION PHI(DMY)                                  PHI        2
                                                                    PHI        3
      C**   PHI COMPUTES LATITUDE.                                  PHI        4
                                                                    PHI        5
            COMMON /STATE/STATE(23)                                 PHI        6
            EQUIVALENCE (STATE(17),CEN31)                           PHI        7
                                                                    PHI        8
            PHI=ASIN(CEN31)                                         PHI        9
            RETURN                                                  PHI       10
            END                                                     PHI       11
```

```
      SUBROUTINE PLOTTER(NSEGT)                                         PLOTTER     2
CC                                                                      PLOTTER     3
CC     PLOTS THE FOLLOWING GRAPHS USING DISSPLA, A USERS LIBRARY        PLOTTER     4
CC     CREATED 1/22/75:                                                 PLOTTER     5
CC     *  LATITUDE VS. LONGITUDE                                        PLOTTER     6
CC     *  ALTITUDE VS. TIME                                             PLOTTER     7
CC     *  ROLL VS. TIME                                                 PLOTTER     8
CC     *  PITCH VS. TIME                                                PLOTTER     9
CC     *  YAW VS. TIME                                                  PLOTTER    10
CC                                                                      PLOTTER    11
      COMMON /GLON/GLON(1001)                                           PLOTTER    12
      COMMON /GLAT/GLAT(1001)                                           PLOTTER    13
      COMMON /GTIM/GTIM(1001)                                           PLOTTER    14
      COMMON /GALT/GALT(1001)                                           PLOTTER    15
      COMMON /GETX/GETX(1001)                                           PLOTTER    16
      COMMON /GETY/GETY(1001)                                           PLOTTER    17
      COMMON /GETZ/GETZ(1001)                                           PLOTTER    18
      COMMON /NPLOT/NPLTPTS,NPLTSEG(50)                                 PLOTTER    19
                                                                        PLOTTER    20
CC                                                                      PLOTTER    21
CC  INITIALIZE CALCOMP PLOTTER                                          PLOTTER    22
      CALL COMPRS                                                       PLOTTER    23
CC                                                                      PLOTTER    24
CC     **********  BEGIN LATITUDE/LONGITUDE PLOT  **********            PLOTTER    25
CC                                                                      PLOTTER    26
CC  INITIALIZE DISSPLA COMMON AREA                                      PLOTTER    27
      CALL BGNPL(1)                                                     PLOTTER    28
CC                                                                      PLOTTER    29
CC   ROTATE PLOT 90 DEGREES AND TRANSLATE                               PLOTTER    30
      CALL BANGLE(-90.)                                                 PLOTTER    31
      CALL BSHIFT(0.,6.)                                                PLOTTER    32
CC                                                                      PLOTTER    33
CC  DETERMINE MAXIMUM AND MINIMUM VALUES                                PLOTTER    34
      CALL MAXMIN(GLON,GLAT,XMAX,XMIN,YMAX,YMIN)                        PLOTTER    35
CC                                                                      PLOTTER    36
CC  POSITION PLOT ORIGIN                                                PLOTTER    37
      CALL PHYSOR(1.5,1.0)                                              PLOTTER    38
CC                                                                      PLOTTER    39
CC  ANNOTATE PLOT                                                       PLOTTER    40
      CALL BASALF("STANDARD")                                           PLOTTER    41
      CALL MIXALF("L/CSTD")                                             PLOTTER    42
      CALL TITLE(1H ,-1,"L(ONGITUDE (DEG))$",100,"L(ATITUDE (DEG))$",100 PLOTTER    43
     1,6.,8.)                                                           PLOTTER    44
      CALL HEADIN("L(ATITUDE/)L(ONGITUDE) F(LIGHT) P(ROFILE)$",-100,-3,1 PLOTTER    45
     1)                                                                 PLOTTER    46
CC                                                                      PLOTTER    47
CC  DETERMINE SCALING FACTORS                                           PLOTTER    48
      CALL SCALE(XMAX,XMIN,YMAX,YMIN,XSTEP,YSTEP,XLEN,YLEN)             PLOTTER    49
      IF (((XLEN/YLEN).GT.6.).OR.((YLEN/XLEN).GT.6.)) GO TO 30          PLOTTER    50
      IF (XSTEP.GT.YSTEP) YSTEP = XSTEP                                 PLOTTER    51
      IF (YSTEP.GT.XSTEP) XSTEP = YSTEP                                 PLOTTER    52
30    CONTINUE                                                          PLOTTER    53
CC                                                                      PLOTTER    54
CC  DRAW FRAME TO ENHANCE PLOT                                          PLOTTER    55
      CALL FRAME                                                        PLOTTER    56
CC                                                                      PLOTTER    57
CC  SET UP GRAPH                                                        PLOTTER    58
```

```
           CALL GRAPH(XMIN,XSTEP,YMIN,YSTEP)                                     PLOTTER    59
       CC                                                                        PLOTTER    60
       CC  ANNOTATE START AND FINISH OF FLIGHT                                   PLOTTER    61
           CALL HEIGHT(0.05)                                                     PLOTTER    62
           CALL RLMESS("*START$",100,GLON(1),GLAT(1))                            PLOTTER    63
           CALL RLMESS("*FINISH$",100,GLON(NPLTPTS),GLAT(NPLTPTS))               PLOTTER    64
       CC                                                                        PLOTTER    65
       CC  MARK FLIGHT SEGMENTS                                                  PLOTTER    66
           CALL HEIGHT(0.06)                                                     PLOTTER    67
           DO 40 I=2,NSEGT                                                       PLOTTER    68
           IM = I                                                                PLOTTER    69
           ENCODE(4,1200,LABEL)IM                                                PLOTTER    70
           MM = NPLTSEG(I)                                                       PLOTTER    71
           CALL RLMESS(LABEL,100,GLON(MM),GLAT(MM))                              PLOTTER    72
       40  CONTINUE                                                              PLOTTER    73
           CALL RESET("HEIGHT")                                                  PLOTTER    74
       CC                                                                        PLOTTER    75
       CC   DRAW DASHED COASTAL OUTLINE ON GRAPH                                 PLOTTER    76
           CALL DASH                                                             PLOTTER    77
           CALL MAPDTA                                                           PLOTTER    78
           CALL RESET("DASH")                                                    PLOTTER    79
       CC                                                                        PLOTTER    80
       CC  DRAW CURVE                                                            PLOTTER    81
           CALL CURVE(GLON,GLAT,NPLTPTS,0)                                       PLOTTER    82
       CC                                                                        PLOTTER    83
       CC  END LATITUDE/LONGITUDE PLOT                                           PLOTTER    84
           CALL ENDPL(1)                                                         PLOTTER    85
       CC                                                                        PLOTTER    86
       CC     **********  BEGIN ALTITUDE PLOT  **********                        PLOTTER    87
       CC                                                                        PLOTTER    88
       CC  INITIALIZE DISSPLA COMMON AREA                                        PLOTTER    89
           CALL BGNPL(2)                                                         PLOTTER    90
       CC                                                                        PLOTTER    91
       CC   ROTATE PLOT 90 DEGREES AND TRANSLATE                                 PLOTTER    92
           CALL BANGLE(-90.)                                                     PLOTTER    93
           CALL BSHIFT(0.,6.)                                                    PLOTTER    94
       CC                                                                        PLOTTER    95
       CC  DETERMINE MAXIMUM AND MINIMUM VALUES                                  PLOTTER    96
           CALL MAXMIN(GTIM,GALT,XMAX,XMIN,YMAX,YMIN)                            PLOTTER    97
       CC                                                                        PLOTTER    98
       CC  POSITION PLOT ORIGIN                                                  PLOTTER    99
           CALL PHYSOR(1.5,1.0)                                                  PLOTTER   100
       CC                                                                        PLOTTER   101
       CC  ANNOTATE PLOT                                                         PLOTTER   102
           CALL BASALF("STANDARD")                                               PLOTTER   103
           CALL MIXALF("L/CSTD")                                                 PLOTTER   104
           CALL TITLE(1H ,-1,"T(IME (SEC))$",100,"A(LTITUDE (FEET))$",100,6., PLOTTER   105
          18.)                                                                   PLOTTER   106
           CALL HEADIN("A(LTITUDE) F(LIGHT) P(ROFILE)$",-100,-3,1)               PLOTTER   107
       CC                                                                        PLOTTER   108
       CC  DETERMINE SCALING FACTORS                                             PLOTTER   109
           CALL SCALE(XMAX,XMIN,YMAX,YMIN,XSTEP,YSTEP,XLEN,YLEN)                 PLOTTER   110
       CC                                                                        PLOTTER   111
       CC  DRAW FRAME TO ENHANCE PLOT                                            PLOTTER   112
           CALL FRAME                                                            PLOTTER   113
       CC                                                                        PLOTTER   114
       CC  SET UP GRAPH                                                          PLOTTER   115
```

```
                CALL GRAPH(XMIN,XSTEP,YMIN,YSTEP)                              PLOTTER   116
          CC                                                                   PLOTTER   117
          CC                                                                   PLOTTER   118
          CC  MARK FLIGHT SEGMENTS                                             PLOTTER   119
                CALL HEIGHT(0.06)                                              PLOTTER   120
                DO 50 I=2,NSEGT                                                PLOTTER   121
                IM = I                                                         PLOTTER   122
                ENCODE(4,1200,LABEL) IM                                        PLOTTER   123
                MM = NPLTSEG(I)                                                PLOTTER   124
                CALL RLMESS(LABEL,100,GTIM(MM),GALT(MM))                       PLOTTER   125
          50    CONTINUE                                                       PLOTTER   126
                CALL RESET ("HEIGHT")                                          PLOTTER   127
          CC                                                                   PLOTTER   128
          CC  DRAW CURVE                                                       PLOTTER   129
                CALL CURVE(GTIM,GALT,NPLTPTS,0)                                PLOTTER   130
          CC                                                                   PLOTTER   131
          CC  END ALTITUDE PLOT                                                PLOTTER   132
                CALL ENDPL(2)                                                  PLOTTER   133
          CC                                                                   PLOTTER   134
          CC   **********  BEGIN ROLL PLOT  **********                         PLOTTER   135
          CC                                                                   PLOTTER   136
          CC  INITIALIZE DISSPLA COMMON AREA                                   PLOTTER   137
                CALL BGNPL(3)                                                  PLOTTER   138
          CC                                                                   PLOTTER   139
          CC   ROTATE PLOT 90 DEGREES AND TRANSLATE                            PLOTTER   140
                CALL BANGLE(-90.)                                              PLOTTER   141
                CALL BSHIFT(0.,6.)                                             PLOTTER   142
          CC                                                                   PLOTTER   143
          CC  DETERMINE MAXIMUM AND MINIMUM VALUES                             PLOTTER   144
                CALL MAXMIN(GTIM,GETX,XMAX,XMIN,YMAX,YMIN)                     PLOTTER   145
          CC                                                                   PLOTTER   146
          CC  POSITION PLOT ORIGIN                                             PLOTTER   147
                CALL PHYSOR(1.5,1.0)                                           PLOTTER   148
          CC                                                                   PLOTTER   149
          CC  ANNOTATE PLOT                                                    PLOTTER   150
                CALL BASALF("STANDARD")                                        PLOTTER   151
                CALL MIXALF("L/CSTD")                                          PLOTTER   152
                CALL TITLE(1H ,-1,"T(IME (SEC))$",100,"R(OLL (DEG))$",100,6.,8.)  PLOTTER   153
                CALL HEADIN("R(OLL) F(LIGHT) P(ROFILE)$",-100,-3,1)           PLOTTER   154
          CC                                                                   PLOTTER   155
          CC  DETERMINE SCALING FACTORS                                        PLOTTER   156
                CALL SCALE(XMAX,XMIN,YMAX,YMIN,XSTEP,YSTEP,XLEN,YLEN)          PLOTTER   157
          CC                                                                   PLOTTER   158
          CC  DRAW FRAME TO ENHANCE PLOT                                       PLOTTER   159
                CALL FRAME                                                     PLOTTER   160
          CC                                                                   PLOTTER   161
          CC  SET UP GRAPH                                                     PLOTTER   162
                CALL GRAPH(XMIN,XSTEP,YMIN,YSTEP)                              PLOTTER   163
          CC                                                                   PLOTTER   164
          CC                                                                   PLOTTER   165
          CC  MARK FLIGHT SEGMENTS                                             PLOTTER   166
                CALL HEIGHT(0.06)                                              PLOTTER   167
                DO 60 I=2,NSEGT                                                PLOTTER   168
                IM = I                                                         PLOTTER   169
                ENCODE(4,1200,LABEL) IM                                        PLOTTER   170
                MM = NPLTSEG(I)                                                PLOTTER   171
                CALL RLMESS(LABEL,100,GTIM(MM),GETX(MM))                       PLOTTER   172
```

```
   60   CONTINUE                                                     PLOTTER   173
        CALL RESET ("HEIGHT")                                        PLOTTER   174
     CC                                                              PLOTTER   175
     CC  DRAW CURVE                                                  PLOTTER   176
        CALL CURVE(GTIM,GETX,NPLTPTS,0)                              PLOTTER   177
     CC                                                              PLOTTER   178
     CC  END ROLL PLOT                                               PLOTTER   179
        CALL ENDPL(3)                                                PLOTTER   180
     CC                                                              PLOTTER   181
     CC    **********  BEGIN PITCH PLOT  **********                  PLOTTER   182
     CC                                                              PLOTTER   183
     CC  INITIALIZE DISSPLA COMMON AREA                              PLOTTER   184
        CALL BGNPL(4)                                                PLOTTER   185
     CC                                                              PLOTTER   186
     CC   ROTATE PLOT 90 DEGREES AND TRANSLATE                       PLOTTER   187
        CALL BANGLE(-90.)                                            PLOTTER   188
        CALL BSHIFT(0.,6.)                                           PLOTTER   189
     CC                                                              PLOTTER   190
     CC  DETERMINE MAXIMUM AND MINIMUM VALUES                        PLOTTER   191
        CALL MAXMIN(GTIM,GETY,XMAX,XMIN,YMAX,YMIN)                   PLOTTER   192
     CC                                                              PLOTTER   193
     CC  POSITION PLOT ORIGIN                                        PLOTTER   194
        CALL PHYSOR(1.5,1.0)                                         PLOTTER   195
     CC                                                              PLOTTER   196
     CC  ANNOTATE PLOT                                               PLOTTER   197
        CALL BASALF("STANDARD")                                      PLOTTER   198
        CALL MIXALF("L/CSTD")                                        PLOTTER   199
        CALL TITLE(1H ,-1,"T(IME (SEC))$",100,"P(ITCH (DEG))$",100,6.,8.) PLOTTER   200
        CALL HEADIN("P(ITCH) F(LIGHT) P(ROFILE)$",-100,-3,1)         PLOTTER   201
     CC                                                              PLOTTER   202
     CC  DETERMINE SCALING FACTORS                                   PLOTTER   203
        CALL SCALE(XMAX,XMIN,YMAX,YMIN,XSTEP,YSTEP,XLEN,YLEN)        PLOTTER   204
     CC                                                              PLOTTER   205
     CC  DRAW FRAME TO ENHANCE PLOT                                  PLOTTER   206
        CALL FRAME                                                   PLOTTER   207
     CC                                                              PLOTTER   208
     CC  SET UP GRAPH                                                PLOTTER   209
        CALL GRAPH(XMIN,XSTEP,YMIN,YSTEP)                            PLOTTER   210
     CC                                                              PLOTTER   211
     CC                                                              PLOTTER   212
     CC  MARK FLIGHT SEGMENTS                                        PLOTTER   213
        CALL HEIGHT(0.06)                                            PLOTTER   214
        DO 70 I=2,NSEGT                                              PLOTTER   215
        IM = I                                                       PLOTTER   216
        ENCODE(4,1200,LABEL) IM                                      PLOTTER   217
        MM = NPLTSEG(I)                                              PLOTTER   218
        CALL RLMESS(LABEL,100,GTIM(MM),GETY(MM))                     PLOTTER   219
   70   CONTINUE                                                     PLOTTER   220
        CALL RESET ("HEIGHT")                                        PLOTTER   221
     CC                                                              PLOTTER   222
     CC  DRAW CURVE                                                  PLOTTER   223
        CALL CURVE(GTIM,GETY,NPLTPTS,0)                              PLOTTER   224
     CC                                                              PLOTTER   225
     CC  END PITCH PLOT                                              PLOTTER   226
        CALL ENDPL(4)                                                PLOTTER   227
     CC                                                              PLOTTER   228
     CC    **********  BEGIN YAW PLOT  **********                    PLOTTER   229
```

```
      CC
      CC  INITIALIZE DISSPLA COMMON AREA
            CALL BGNPL(5)
      CC
      CC   ROTATE PLOT 90 DEGREES AND TRANSLATE
            CALL EANGLE(-90.)
            CALL BSHIFT(0.,6.)
      CC
      CC  DETERMINE MAXIMUM AND MINIMUM VALUES
            CALL MAXMIN(GTIM,GETZ,XMAX,XMIN,YMAX,YMIN)
      CC
      CC  POSITION PLOT ORIGIN
            CALL PHYSOR(1.5,1.0)
      CC
      CC  ANNOTATE PLOT
            CALL BASALF("STANDARD")
            CALL MIXALF("L/CSTD")
            CALL TITLE(1H ,-1,"T(IME (SEC))$",100,"Y(AW (DEG))$",100,6.,8.)
            CALL HEADIN("Y(AW) F(LIGHT) P(ROFILE)$",-100,-3,1)
      CC
      CC  DETERMINE SCALING FACTORS
            CALL SCALE(XMAX,XMIN,YMAX,YMIN,XSTEP,YSTEP,XLEN,YLEN)
      CC
      CC  DRAW FRAME TO ENHANCE PLOT
            CALL FRAME
      CC
      CC  SET UP GRAPH
            CALL GRAPH(XMIN,XSTEP,YMIN,YSTEP)
      CC
      CC
      CC  MARK FLIGHT SEGMENTS
            CALL HEIGHT(0.06)
            DO 80 I=2,NSEGT
            IM = I
            ENCODE(4,1200,LABEL) IM
            MM = NPLTSEG(I)
            CALL RLMESS(LABEL,100,GTIM(MM),GETZ(MM))
      80    CONTINUE
            CALL RESET ("HEIGHT")
      CC
      CC  DRAW CURVE
            CALL CURVE(GTIM,GETZ,NPLTPTS,0)
      CC
      CC  END YAW PLOT
            CALL ENDPL(5)
      CC
      CC  SIGNAL DISSPLA TO TERMINATE THE PLOT
      100   CALL DONEPL
      1200  FORMAT(I3,"$")
            END
```

```
      SUBROUTINE PRNTOUT                                            PRNTOUT    2
                                                                    PRNTOUT    3
C**  PRNTOUT PRINTS OUTPUT IN G FORMAT AND                          PRNTOUT    4
C**  LABELS EACH OUTPUT VARIABLE.                                   PRNTOUT    5
                                                                    PRNTOUT    6
      COMMON /FIXED/FIXED(15)                                       PRNTOUT    7
      COMMON /PACC/PACC(50)                                         PRNTOUT    8
      COMMON /STATE/X(23)                                           PRNTOUT    9
      COMMON /SUPLE/SUPLE(9)                                        PRNTOUT   10
                                                                    PRNTOUT   11
      EQUIVALENCE (FIXED(2),RADPERD)                                PRNTOUT   12
      EQUIVALENCE (X(1),VX)                                         PRNTOUT   13
      EQUIVALENCE (X(2),VY)                                         PRNTOUT   14
      EQUIVALENCE (X(3),VZ)                                         PRNTOUT   15
      EQUIVALENCE (X(4),VT)                                         PRNTOUT   16
      EQUIVALENCE (X(5),ALT)                                        PRNTOUT   17
      EQUIVALENCE (SUPLE(1),T)                                      PRNTOUT   18
      EQUIVALENCE (SUPLE(3),TI)                                     PRNTOUT   19
      EQUIVALENCE (SUPLE(6),ISEG)                                   PRNTOUT   20
                                                                    PRNTOUT   21
      REAL LAMDA                                                    PRNTOUT   22
                                                                    PRNTOUT   23
      TOUTNEW=TOUT(DMY)                                             PRNTOUT   24
      IF (T.EQ.TI) GO TO 10                                         PRNTOUT   25
      IF (TOUTOLD.EQ.TOUTNEW) RETURN                                PRNTOUT   26
   10 OPHI=PHI(DMY)/RADPERD                                         PRNTOUT   27
      OLAMDA=LAMDA(DMY)/RADPERD                                     PRNTOUT   28
      OALFA=ALFA(DMY)/RADPERD                                       PRNTOUT   29
      OETAX=ETAX(DMY)/RADPERD                                       PRNTOUT   30
      OETAY=ETAY(DMY)/RADPERD                                       PRNTOUT   31
      OETAZ=ETAZ(DMY)/RADPERD                                       PRNTOUT   32
      OPSI=PSI(DMY)/RADPERD                                         PRNTOUT   33
      CALL ACCLRTN(FX,FY,FZ)                                        PRNTOUT   34
      CALL ETADOT(ETAXDOT,ETAYDOT,ETAZDOT)                          PRNTOUT   35
      ETAXDOT=ETAXDOT/RADPERD                                       PRNTOUT   36
      ETAYDOT=ETAYDOT/RADPERD                                       PRNTOUT   37
      ETAZDOT=ETAZDOT/RADPERD                                       PRNTOUT   38
      WRITE (6,100) T,                                              PRNTOUT   39
     1            OPHI,         OLAMDA,       OALFA,        ALT,    PRNTOUT   40
     2            OETAX,        OETAY,        OETAZ,        OPSI,   PRNTOUT   41
     3            ETAXDOT,      ETAYDOT,      ETAZDOT,              PRNTOUT   42
     4            VX,           VY,           VZ,           VT,     PRNTOUT   43
     5            FX,           FY,           FZ,           PACC(ISEG) PRNTOUT 44
  100 FORMAT(//,T3,*TIME*,T9,F12.5/T8,*LAT*,T13,G20.10,T37,*LON*,T43, PRNTOUT 45
     A        G20.10,T67,*ALPHA*,T73,G20.10,T97,*ALT*,T103,G20.10/  PRNTOUT   46
     B        T8,*ROLL*,T13,G20.10,T37,*PITCH*,T43,G20.10,T67,*YAW*, PRNTOUT  47
     C        T73,G20.10,T97,*PSI*,T103,G20.10/                     PRNTOUT   48
     D        T8,*DROLL*,T13,G20.10,T37,*DPITCH*,T43,G20.10,T67,*DYAW*, PRNTOUT 49
     E        T73,G20.10/                                           PRNTOUT   50
     F        T8,*VX*,T13,G20.10,T37,*VY*,T43,G20.10,T67,*VZ*,T73,  PRNTOUT   51
     G        G20.10,T97,*VPATH*,T103,G20.10/                       PRNTOUT   52
     H        T8,*FX*,T13,G20.10,T37,*FY*,T43,G20.10,T67,*FZ*,T73,  PRNTOUT   53
     I        G20.10,T97,*APATH*,T103,G20.10 )                      PRNTOUT   54
      TOUTOLD=TOUTNEW                                               PRNTOUT   55
      RETURN                                                        PRNTOUT   56
      END                                                           PRNTOUT   57
```

```
      REAL FUNCTION PSI(DMY)                                            PSI   2
                                                                        PSI   3
C**   PSI COMPUTES THE GROUND TRACK HEADING ANGLE WHICH IS MEASURED     PSI   4
C**   POSITIVE CW FROM NORTH. THE INITIAL VALUE OF PSI IS PHEADO.       PSI   5
C**   PSI'S RANGE IS (-PI,+PI).                                         PSI   6
                                                                        PSI   7
      COMMON /FIXED/FIXED(15)                                           PSI   8
      EQUIVALENCE (FIXED(3),TWOPI)                                      PSI   9
      EQUIVALENCE (FIXED(4),PI)                                         PSI  10
                                                                        PSI  11
      PSI=ETAZ(DMY)-ALFA(DMY)                                           PSI  12
      IF (PSI.LT.-PI) PSI=PSI+TWOPI                                     PSI  13
      IF (PSI.GT. PI) PSI=PSI-TWOPI                                     PSI  14
      RETURN                                                            PSI  15
      END                                                               PSI  16
```

```
      REAL FUNCTION PSIDOT(OMY)                                        PSIDOT   2
                                                                       PSIDOT   3
C**   PSIDOT COMPUTES THE ANGULAR RATE-OF-CHANGE OF AIRCRAFT HEADING.  PSIDOT   4
                                                                       PSIDOT   5
      COMMON /NPATH/NPATH(50)                                          PSIDOT   6
      COMMON /STATE/STATE(23)                                          PSIDOT   7
      COMMON /SUPLE/SUPLE(9)                                           PSIDOT   8
      COMMON /TURN/TURN(50)                                            PSIDOT   9
                                                                       PSIDOT  10
      EQUIVALENCE (STATE(4),VT)                                        PSIDOT  11
      EQUIVALENCE (SUPLE(4),TRNDONE)                                   PSIDOT  12
      EQUIVALENCE (SUPLE(6),ISEG)                                      PSIDOT  13
                                                                       PSIDOT  14
      INTEGER TURN                                                     PSIDOT  15
                                                                       PSIDOT  16
C                                                                      PSIDOT  17
C             GREAT CIRCLE CONTRIBUTION                                PSIDOT  18
      PSIDOT=0.                                                        PSIDOT  19
      IF (NPATH(ISEG).EQ.1) PSIDOT=PSIDOTG(OMY)                        PSIDOT  20
      GO TO (19,20,30,10) TURN(ISEG)                                   PSIDOT  21
C                                                                      PSIDOT  22
C             VERTICAL TURNS AND STRAIGHT FLIGHT PATHS                 PSIDOT  23
   10 RETURN                                                           PSIDOT  24
C                                                                      PSIDOT  25
C             HORIZONTAL TURNS                                         PSIDOT  26
   20 PSIDOT=PSIDOT+32.2*TAN(ETAX(OMY))*(1.-TRNDONE)/VT                PSIDOT  27
      RETURN                                                           PSIDOT  28
C                                                                      PSIDOT  29
C             SINE HEADING CHANGES                                     PSIDOT  30
   30 PSIDOT=PSIDOT+32.2*TAN(ETAX(OMY))/VT                             PSIDOT  31
      RETURN                                                           PSIDOT  32
      END                                                              PSIDOT  33
```

```
      REAL FUNCTION PSIDOTG(DMY)
C
C**   PSIDOTG COMPUTES THE PSI RATE REQUIRED TO
C**   KEEP THE AIRCRAFT IN A GREAT CIRCLE PLANE.
C**   REMARK:  FOR A SPHERICAL EARTH (ESQ=0), THE CORRECT PSI RATE
C**   IS LAMDOT*SIN(PHI), A FACT THAT CAN BE DEDUCED FROM THE EQUATIONS
C**   BELOW.  USING LAMDOT*SIN(PHI) OVER AN ELLIPSOIDAL EARTH PRODUCES
C**   SMALL TURNING-RATE ERRORS THAT CAUSE THE AIRCRAFT TO STRAY FROM
C**   THE GREAT CIRCLE PLANE.  IF SMALL GREAT CIRCLE ERRORS ARE TOLER-
C**   ABLE, THE USER MAY WISH TO SUBSTITUTE LAMDOT*SIN(PHI) FOR THE
C**   EQUATIONS BELOW TO SAVE COMPUTING TIME.

      COMMON /FIXED/FIXED(15)
      COMMON /STATE/STATE(23)

      EQUIVALENCE (FIXED(6),RE)
      EQUIVALENCE (FIXED(7),ESQ)
      EQUIVALENCE (STATE(1),VX)
      EQUIVALENCE (STATE(2),VY)
      EQUIVALENCE (STATE(3),HDOT)
      EQUIVALENCE (STATE(5),H)
      EQUIVALENCE (STATE(17),SFI)

      REAL LAMDOT
C
C         TRIG FUNCTIONS OF PSI, PHI AND DV (GEOGRAPHIC
C         MINUS GEOCENTRIC LATITUDE)
      SI=PSI(DMY)
      SSI=SIN(SI)
      IF (SSI.EQ.0.) PSIDOTG=0.
      IF (SSI.EQ.0.) RETURN
      CSI=COS(SI)
      CSISQ=CSI*CSI
      FI=PHI(DMY)
      CFI=COS(FI)
      SFISQ=SFI*SFI
      CFISQ=CFI*CFI
      B1=1.-ESQ*SFISQ
      B2=SQRT(B1)
      B3=H/RE
      R=RE*SQRT((1.-(2.-ESQ)*ESQ*SFISQ)/B1+2.*B2*B3+B3*B3)
      SDV=(RE*ESQ*SFI*CFI/(R*B2))
      DV=ASIN(SDV)
      CDV=COS(DV)
      TDV=SDV/CDV
C
C         DERIVATIVES OF PHI, R AND DV
      ALPHA=ALFA(DMY)
      VNORTH=VX*COS(ALPHA)-VY*SIN(ALPHA)
      FIDOT=VNORTH/(RM(DMY)+H)
      RDOT=(RE/R)*(HDOT*(B2+B3)-RE*ESQ*SFI*CFI*FIDOT*
     1    ((1.-ESQ)/(B1*B1)+B3/B2))
      DVDOT=(RE*ESQ/(R*B2*CDV))*((CFISQ/B1-SFISQ)*FIDOT-RDOT*SFI*CFI/R)
C
C         DERIVATIVE OF PSI FOR GREAT CIRCLE FLIGHT
      PSIDOTG=LAMDOT(DMY)*SIN(FI-DV)*CDV*(1.+CSISQ*TDV*TDV) +
     1        DVDOT*SSI*CSI*TDV
```

```
      RETURN                                                   PSIDOTG    59
      END                                                      PSIDOTG    60
```

```
      SUBROUTINE QUADRT(A,B,C,IWARN,XR,XI)
C
C**   QUADRT SOLVES THE QUADRATIC EQUATION
C**         A X**2 + B X + C = 0
C**   A, B AND C MUST BE REAL. ON RETURN, THE
C**   TWO ROOTS ARE AVAILABLE AS FOLLOWS:
C**         X1 = XR(1) + XI(1)*SQRT(-1)
C**         X2 = XR(2) + XI(2)*SQRT(-1)
C**   IWARN IS SET TO 1 IF ROOTS DON'T EXIST OR IF
C**   THE EXISTING ROOTS HAVE NONZERO IMAGINARY PARTS.
C
      LOGICAL BPOS, COANEG
      DIMENSION XR(2), XI(2)
C
      IWARN = 0
      IF(A.EQ.0.) GO TO 40
C
C**           TWO ROOTS (A NOT EQUAL 0)
C
      IF(B.EQ.0.) GO TO 10
      BPOS = .TRUE.
      IF(B.LT.0.) BPOS = .FALSE.
      D = B*B-4.*A*C
      IF(D.EQ.0.) GO TO 20
      IF(D.LT.0.) GO TO 30
C
C             TWO REAL UNEQUAL ROOTS (D>0 AND B NOT 0)
      XI(1) = XI(2) = 0.
      D = SQRT(D)
      IF(BPOS) XR(1) = -2.*C/(B + D)
      IF(BPOS) XR(2) = -(B + D)/(2.*A)
      IF(BPOS) RETURN
      XR(1) = (-B + D)/(2.*A)
      XR(2) = 2.*C/(-B+D)
      RETURN
C
C             TWO ROOTS OF EQUAL MAGNITUDE BUT OPPOSITE SIGN (B=0)
   10 COANEG = .TRUE.
      COA = C/A
      IF(COA.GE.0.) COANEG = .FALSE.
      IF(COANEG) XR(1) = SQRT(-COA)
      IF(COANEG) XR(2) = -XR(1)
      IF(COANEG) XI(1) = XI(2) = 0.
      IF(COANEG) RETURN
      XR(1) = XR(2) = 0.
      XI(1) = SQRT(COA)
      XI(2) = -XI(1)
      IWARN = 1
      RETURN
C
C             TWO REAL EQUAL ROOTS (D=0 AND B NOT 0)
   20 XI(1) = XI(2) = 0.
      XR(1) = XR(2) = -B/(2*A)
      RETURN
C
C             TWO IMAGINARY ROOTS (D<0 AND B NOT 0)
   30 D = SQRT(-D)
```

```
SUBROUTINE QUADRT     74/74    OPT=2                          FTN 4.5+414       06/11/76  13.22.47        PAGE    4

            XI(1) = D/(2.*A)                                                    QUADRT      59
            XI(2) = -XI(1)                                                      QUADRT      60
            XR(1) = XR(2) = -B/(2.*A)                                           QUADRT      61
            IWARN = 1                                                           QUADRT      62
            RETURN                                                              QUADRT      63
       40   IF(B.EQ.0.) GO TO 50                                                QUADRT      64
      C                                                                         QUADRT      65
      C**          ONE REAL ROOT (A=0 BUT B NOT 0)                              QUADRT      66
      C                                                                         QUADRT      67
            XR(1) = -C/B                                                        QUADRT      68
            XR(2) = XI(1) = XI(2) = 0.                                          QUADRT      69
            RETURN                                                              QUADRT      70
      C                                                                         QUADRT      71
      C**          NO ROOTS (A AND B BOTH ZERO)                                 QUADRT      72
      C                                                                         QUADRT      73
       50   XR(1) = XR(2) = XI(1) = XI(2) = 0.                                  QUADRT      74
            IWARN = 1                                                           QUADRT      75
            WRITE (6,100)                                                       QUADRT      76
            RETURN                                                              QUADRT      77
      100   FORMAT(T2,"QUADRT MESSAGE -  SINCE A AND B ARE ZERO, NO SOLUTIONS  QUADRT      78
           1COULD BE FOUND.")                                                   QUADRT      79
            END                                                                 QUADRT      80
```

```
      SUBROUTINE RHONE(RHO)

C**   RHONE COMPUTES THE ANGULAR RATES OF THE NAV FRAME WITH RESPECT TO
C**   THE EARTH FRAME. THESE RATES ARE COORDINATIZED IN THE NAV FRAME.

      COMMON /FIXED/FIXED(15)
      COMMON /PRBLK/PRBLK(12)
      COMMON /STATE/STATE(23)

      EQUIVALENCE (FIXED(8),WEI)
      EQUIVALENCE (PRBLK(1),LLMECH)
      EQUIVALENCE (STATE(1),VX)
      EQUIVALENCE (STATE(2),VY)
      EQUIVALENCE (STATE(5),ALT)
      EQUIVALENCE (STATE(17),SINEPHI)

      REAL J,LAMDOT
      DIMENSION RHO(3)

      A=ALFA(DMY)
      CA=COS(A)
      SA=SIN(A)
      VWEST=-VEAST(DMY)
      VNORTH=VX*CA-VY*SA
      RHOWEST=VNORTH/(RM(DMY)+ALT)
      RHONRTH=-VWEST/(RP(DMY)+ALT)
      RHO(1)= RHONRTH*CA+RHOWEST*SA
      RHO(2)=-RHONRTH*SA+RHOWEST*CA
      GO TO (10,20,30,40) LLMECH
   10 RHO(3)=0.
      RETURN
   20 RHO(3)=LAMDOT(DMY)*SINEPHI
      RETURN
   30 J=SIGN(1.,PHI(DMY))
      RHO(3)=LAMDOT(DMY)*(SINEPHI-J)
      RETURN
   40 RHO(3)=-WEI*SINEPHI
      RETURN
      END
```

```
          SUBROUTINE RITEOUT                                               RITEOUT     2
                                                                           RITEOUT     3
C**    RITEOUT WRITES OUTPUT ON TAPE3 WITH NO FORMAT CONVERSION.           RITEOUT     4
C**    EACH CALL TO RITEOUT CREATES ONE BINARY RECORD.                     RITEOUT     5
                                                                           RITEOUT     6
          COMMON /PRBLK/PRBLK(13)                                          RITEOUT     7
          COMMON /STATE/X(23)                                              RITEOUT     8
          COMMON /SUPLE/SUPLE(6)                                           RITEOUT     9
                                                                           RITEOUT    10
          EQUIVALENCE (PRBLK(2),TSTART)                                    RITEOUT    11
          EQUIVALENCE (X(1),VX)                                            RITEOUT    12
          EQUIVALENCE (X(2),VY)                                            RITEOUT    13
          EQUIVALENCE (X(3),VZ)                                            RITEOUT    14
          EQUIVALENCE (X(5),ALT)                                           RITEOUT    15
          EQUIVALENCE (SUPLE(1),T)                                         RITEOUT    16
                                                                           RITEOUT    17
          REAL LAMDA                                                       RITEOUT    18
                                                                           RITEOUT    19
          TOUTNEW=TOUT(DMY)                                                RITEOUT    20
          IF (T.EQ.TSTART) GO TO 10                                        RITEOUT    21
          IF (TOUTOLD.EQ.TOUTNEW) RETURN                                   RITEOUT    22
       10 CALL ACCLRTN(FX,FY,FZ)                                           RITEOUT    23
          WRITE (3) T,PHI(DMY),LAMDA(DMY),ALFA(DMY),ALT,FTAX(DMY),FTAY(DMY), RITEOUT  24
         1          FTAZ(DMY),VX,VY,VZ,FX,FY,FZ                            RITEOUT    25
          TOUTOLD=TOUTNEW                                                  RITEOUT    26
          RETURN                                                           RITEOUT    27
          END                                                              RITEOUT    28
```

```
      REAL FUNCTION RM(DMY)                                   RM    2
                                                              RM    3
C**   RM COMPUTES THE RADIUS OF CURVATURE OF AN EARTH         RM    4
C**   MERIDIAN LINE AT LATITUDE PHI.                          RM    5
                                                              RM    6
      COMMON /FIXED/FIXED(15)                                 RM    7
      COMMON /STATE/STATE(23)                                 RM    8
                                                              RM    9
      EQUIVALENCE (FIXED(6),RE)                               RM   10
      EQUIVALENCE (FIXED(7),ESQ)                              RM   11
      EQUIVALENCE (STATE(17),SINEPHI)                         RM   12
                                                              RM   13
      RM=RE*(1.-ESQ)/(1.-ESQ*SINEPHI*SINEPHI)**1.5            RM   14
      RETURN                                                  RM   15
      END                                                     RM   16
```

```
      REAL FUNCTION ROLDOTC(ITURN)                                      ROLDOTC   2
                                                                        ROLDOTC   3
C**  ROLDOTC COMPUTES COMMANDED ROLL RATE FOR BOTH                      ROLDOTC   4
C**  HORIZONTAL TURNS AND SINE HEADING CHANGES.                         ROLDOTC   5
                                                                        ROLDOTC   6
      COMMON /HEAD/HEAD(50)                                             ROLDOTC   7
      COMMON /PITCH/PITCH(50)                                           ROLDOTC   8
      COMMON /PRBLK/PRBLK(13)                                           ROLDOTC   9
      COMMON /STATE/STATE(23)                                           ROLDOTC  10
      COMMON /SUPLE/SUPLE(9)                                            ROLDOTC  11
                                                                        ROLDOTC  12
      EQUIVALENCE (PRBLK(13),ROLRATE)                                   ROLDOTC  13
      EQUIVALENCE (STATE(4),VT)                                         ROLDOTC  14
      EQUIVALENCE (SUPLE(1),T)                                          ROLDOTC  15
      EQUIVALENCE (SUPLE(3),TI)                                         ROLDOTC  16
      EQUIVALENCE (SUPLE(6),ISEG)                                       ROLDOTC  17
      EQUIVALENCE (SUPLE(9),RRCOEF)                                     ROLDOTC  18
                                                                        ROLDOTC  19
      IF (ITURN.EQ.3) GO TO 10                                          ROLDOTC  20
C                                                                       ROLDOTC  21
C**  ROLL RATE COMMAND FOR A HORIZONTAL TURN                            ROLDOTC  22
C                                                                       ROLDOTC  23
      RTLF=SIGN(1.,HEAD(ISEG))                                          ROLDOTC  24
      ROLDOTC=ROLRATE*RTLF*RRCOEF                                       ROLDOTC  25
      RETURN                                                            ROLDOTC  26
C                                                                       ROLDOTC  27
C**  ROLL RATE COMMAND FOR A SINE MANEUVER                              ROLDOTC  28
C                                                                       ROLDOTC  29
   10 TWOWT=2.*PITCH(ISEG)*(T-TI)                                       ROLDOTC  30
      WA=PITCH(ISEG)*HEAD(ISEG)                                         ROLDOTC  31
      SIDOT1=RRCOEF*WA*SIN(TWOWT)                                       ROLDOTC  32
      SIDOT2=RRCOEF*2.*WA*PITCH(ISEG)*COS(TWOWT)                        ROLDOTC  33
      ROLDOTC=32.2*VT*SIDOT2/(32.2*32.2+VT*VT*SIDOT1*SIDOT1)            ROLDOTC  34
      ROLDOTC=SIGN(1.,ROLDOTC)*AMIN1(ROLRATE,ABS(ROLDOTC))              ROLDOTC  35
      RETURN                                                            ROLDOTC  36
      END                                                               ROLDOTC  37
```

```
FUNCTION RP          74/74    OPT=2                    FTN 4.5+414        05/11/76  13.22.47        PAGE     1

                REAL FUNCTION RP(DMY)                                          RP         2
                                                                               RP         3
          C**   RP COMPUTES THE RADIUS OF CURVATURE OF THE EARTH ELLIPSOID IN A RP         4
          C**   PLANE THRU THE NORMAL AND AT RIGHT ANGLES TO THE MERIDIAN.     RP         5
                                                                               RP         6
                COMMON /FIXED/FIXED(15)                                        RP         7
                COMMON /STATE/STATE(23)                                        RP         8
                                                                               RP         9
                EQUIVALENCE (FIXED(6),RE)                                      RP        10
                EQUIVALENCE (FIXED(7),ESQ)                                     RP        11
                EQUIVALENCE (STATE(17),SINEPHI)                                RP        12
                                                                               RP        13
                RP=RE/SQRT(1.-ESQ*SINEPHI*SINEPHI)                             RP        14
                RETURN                                                         RP        15
                END                                                            RP        16
```

```
      SUBROUTINE SCALE(XMAX,XMIN,YMAX,YMIN,XSTEP,YSTEP,XLEN,YLEN)    SCALE   2
                                                                     SCALE   3
CC    ROUTINE TO SCALE VARIABLES FOR PLOTTING ROUTINES.              SCALE   4
                                                                     SCALE   5
      IFLAG = 1                                                      SCALE   6
   10 M = 0                                                          SCALE   7
      I=0                                                            SCALE   8
      XLEN = XMAX - XMIN                                             SCALE   9
      YLEN = YMAX - YMIN                                             SCALE  10
      IF (XLEN.EQ.0.) GO TO 140                                      SCALE  11
      XSTEP = XLEN/6.                                                SCALE  12
      IF(XSTEP.EQ.0.) XSTEP = 0.1                                    SCALE  13
      IF (XSTEP.EQ.0.) GO TO 70                                      SCALE  14
      IXSTEP = INT(XSTEP)                                            SCALE  15
   30 IF (IXSTEP.GT.9) GO TO 40                                      SCALE  16
      M = M+1                                                        SCALE  17
      IXSTEP = INT(XSTEP*(10.**M))                                   SCALE  18
      GO TO 30                                                       SCALE  19
   40 IF (IXSTEP.GE.99) GO TO 50                                     SCALE  20
      XSTEP = XSTEP*(10.**M)                                         SCALE  21
      IXSTEP = INT(XSTEP)                                            SCALE  22
      XSTEP = FLOAT(IXSTEP)/(10.**M)                                 SCALE  23
      GO TO 60                                                       SCALE  24
   50 XSTEP = FLOAT(IXSTEP)/(10.**M)                                 SCALE  25
   60 IF ((XSTEP*6.).GE.XLEN) GO TO 70                               SCALE  26
      XSTEP = XSTEP*(10.**M)                                         SCALE  27
      N = N + 1                                                      SCALE  28
      IXSTEP = INT(XSTEP) + N                                        SCALE  29
      XSTEP = FLOAT(IXSTEP)/(10.**M)                                 SCALE  30
      GO TO 60                                                       SCALE  31
   70 CONTINUE                                                       SCALE  32
      MY=0                                                           SCALE  33
      NY=0                                                           SCALE  34
      YSTEP = YLEN/8.                                                SCALE  35
      IF (YSTEP.EQ.0.) YSTEP = 0.1                                   SCALE  36
      IF (YSTEP.EQ.0.1) GO TO 130                                    SCALE  37
      IYSTEP = INT(YSTEP)                                            SCALE  38
   80 IF (IYSTEP.GT.9) GO TO 90                                      SCALE  39
      MY = MY+1                                                      SCALE  40
      IYSTEP = INT(YSTEP*(10.**MY))                                  SCALE  41
      GO TO 80                                                       SCALE  42
   90 IF (IYSTEP.GE.99) GO TO 100                                    SCALE  43
      YSTEP = YSTEP*(10.**MY)                                        SCALE  44
      IYSTEP = INT(YSTEP)                                            SCALE  45
      YSTEP = FLOAT(IYSTEP)/(10.**MY)                                SCALE  46
      GO TO 110                                                      SCALE  47
  100 YSTEP = FLOAT(IYSTEP)/(10.**MY)                                SCALE  48
  110 IF ((YSTEP*8.).GE.YLEN) GO TO 120                              SCALE  49
      YSTEP = YSTEP*(10.**MY)                                        SCALE  50
      NY = NY + 1                                                    SCALE  51
      IYSTEP = INT(YSTEP) + NY                                       SCALE  52
      YSTEP = FLOAT(IYSTEP)/(10.**MY)                                SCALE  53
      GO TO 110                                                      SCALE  54
  120 CONTINUE                                                       SCALE  55
      IF (IFLAG.LE.0) GO TO 130                                      SCALE  56
      XNEGINC = XMIN/XSTEP                                           SCALE  57
      IXNEGIN = INT(XNEGINC)                                         SCALE  58
```

```
      IF ((ABS(AMOD(XMIN,XSTEP))).GT.0.) XMIN = FLOAT(IXNEGIN - 1)*XSTEP SCALE    59
      XLEN = XMAX - XMIN                                                 SCALE    60
      XPOSINC = XMAX/XSTEP                                               SCALE    61
      IXPOSIN = INT(XPOSINC)                                             SCALE    62
      IF ((ABS(AMOD(XLEN,XSTEP))).GT.0.) XMAX = FLOAT(IXPOSIN + 1)*XSTEP SCALE    63
      YNEGINC = YMIN/YSTEP                                               SCALE    64
      IYNEGIN = INT(YNEGINC)                                             SCALE    65
      IF ((ABS(AMOD(YMIN,YSTEP))).GT.0.) YMIN = FLOAT(IYNEGIN - 1)*YSTEP SCALE    66
      YLEN = YMAX - YMIN                                                 SCALE    67
      YPOSINC = YMAX/YSTEP                                               SCALE    68
      IYPOSIN = INT(YPOSINC)                                             SCALE    69
      IF ((ABS(AMOD(YLEN,YSTEP))).GT.0.) YMAX = FLOAT(IYPOSIN + 1)*YSTEP SCALE    70
      IFLAG = IFLAG - 1                                                  SCALE    71
      IF (IFLAG.GE.0) GO TO 10                                           SCALE    72
  130 CONTINUE                                                           SCALE    73
      RETURN                                                             SCALE    74
  140 WRITE(6,1000)                                                      SCALE    75
      CALL EXIT                                                          SCALE    76
 1000 FORMAT(2X"*****  YOU ARE TRYING TO GRAPH A NULL PLOT. PROGRAM WILL SCALE    77
     1 BE TERMINATED  *****")                                            SCALE    78
      END                                                                SCALE    79
```

```
      SUBROUTINE SKEW(A,B)                                    SKEW   2
                                                              SKEW   3
C**   SKEW FORMS THE 3X3 SKEW-SYMMETRIC MATRIX, B,            SKEW   4
C**   CORRESPONDING TO THE 3X1 VECTOR, A.                     SKEW   5
                                                              SKEW   6
      DIMENSION A(3),B(3,3)                                   SKEW   7
                                                              SKEW   8
      B(1,1)=0.0                                              SKEW   9
      B(1,2)=-A(3)                                            SKEW  10
      B(1,3)=A(2)                                             SKEW  11
      B(2,1)=A(3)                                             SKEW  12
      B(2,2)=0.0                                              SKEW  13
      B(2,3)=-A(1)                                            SKEW  14
      B(3,1)=-A(2)                                            SKEW  15
      B(3,2)=A(1)                                             SKEW  16
      B(3,3)=0.0                                              SKEW  17
      RETURN                                                  SKEW  18
      END                                                     SKEW  19
```

```
      SUBROUTINE SVSETUP                                                SVSETUP   2
                                                                        SVSETUP   3
C**   SVSETUP INCORPORATES THE INITIAL PROBLEM DATA INTO THE            SVSETUP   4
C**   STATE VECTOR. THIS IS ALWAYS DONE BEFORE BEGINNING THE FIRST      SVSETUP   5
C**   FLIGHT SEGMENT. IT CAN BE REPEATED AT THE BEGINNING OF OTHER      SVSETUP   6
C**   SEGMENTS IF THE USER WISHES ADDITIONAL FLIGHT PROFILES BEGINNING  SVSETUP   7
C**   FROM THE ORIGONAL STARTING POINT.                                 SVSETUP   8
                                                                        SVSETUP   9
      COMMON /PRBLK/PRBLK(13)                                           SVSETUP  10
      COMMON /STATE/X(23)                                               SVSETUP  11
                                                                        SVSETUP  12
      EQUIVALENCE (PRBLK(3),VTO)                                        SVSETUP  13
      EQUIVALENCE (PRBLK(4),PHEADO)                                     SVSETUP  14
      EQUIVALENCE (PRBLK(5),PPITCHO)                                    SVSETUP  15
      EQUIVALENCE (PRBLK(6),ALFO)                                       SVSETUP  16
      EQUIVALENCE (PRBLK(7),PHIO)                                       SVSETUP  17
      EQUIVALENCE (PRBLK(8),LAMO)                                       SVSETUP  18
      EQUIVALENCE (PRBLK(9),ALTO)                                       SVSETUP  19
      EQUIVALENCE (X(1),V(1),VX)                                        SVSETUP  20
      EQUIVALENCE (X(2),V(2),VY)                                        SVSETUP  21
      EQUIVALENCE (X(3),V(3),VZ)                                        SVSETUP  22
      EQUIVALENCE (X(4),VT)                                             SVSETUP  23
      EQUIVALENCE (X(5),ALT)                                            SVSETUP  24
      EQUIVALENCE (X(6),CPN(1,1))                                       SVSETUP  25
      EQUIVALENCE (X(15),CEN(1,1))                                      SVSETUP  26
                                                                        SVSETUP  27
      REAL LAMO                                                         SVSETUP  28
      DIMENSION CEN(3,3),CPN(3,3),V(3)                                  SVSETUP  29
                                                                        SVSETUP  30
      ETAXO=XO=0.                                                       SVSETUP  31
      ETAYO=YO=PPITCHO                                                  SVSETUP  32
      ETAZO=ZO=ALFO+PHEADO                                              SVSETUP  33
      CPN(1,1)= COS(ZO)*COS(YO)                                         SVSETUP  34
      CPN(2,1)=-SIN(ZO)*COS(YO)                                         SVSETUP  35
      CPN(3,1)= SIN(YO)                                                 SVSETUP  36
      CPN(1,2)= COS(ZO)*SIN(YO)*SIN(XO)-SIN(ZO)*COS(XO)                 SVSETUP  37
      CPN(2,2)=-SIN(ZO)*SIN(YO)*SIN(XO)-COS(ZO)*COS(XO)                 SVSETUP  38
      CPN(3,2)=-COS(YO)*SIN(XO)                                         SVSETUP  39
      CPN(1,3)= COS(ZO)*SIN(YO)*COS(XO)+SIN(ZO)*SIN(XO)                 SVSETUP  40
      CPN(2,3)= COS(ZO)*SIN(XO)-SIN(ZO)*SIN(YO)*COS(XO)                 SVSETUP  41
      CPN(3,3)=-COS(YO)*COS(XO)                                         SVSETUP  42
      CEN(1,1)= COS(ALFO)*COS(PHIO)                                     SVSETUP  43
      CEN(2,1)=-SIN(ALFO)*COS(PHIO)                                     SVSETUP  44
      CEN(3,1)= SIN(PHIO)                                               SVSETUP  45
      CEN(1,2)= SIN(ALFO)*COS(LAMO)+COS(ALFO)*SIN(PHIO)*SIN(LAMO)       SVSETUP  46
      CEN(2,2)= COS(ALFO)*COS(LAMO)-SIN(ALFO)*SIN(PHIO)*SIN(LAMO)       SVSETUP  47
      CEN(3,2)=-COS(PHIO)*SIN(LAMO)                                     SVSETUP  48
      CEN(1,3)= SIN(ALFO)*SIN(LAMO)-COS(ALFO)*SIN(PHIO)*COS(LAMO)       SVSETUP  49
      CEN(2,3)= COS(ALFO)*SIN(LAMO)+SIN(ALFO)*SIN(PHIO)*COS(LAMO)       SVSETUP  50
      CEN(3,3)= COS(PHIO)*COS(LAMO)                                     SVSETUP  51
      V(1)=VTO                                                          SVSETUP  52
      V(2)=V(3)=0.                                                      SVSETUP  53
      CALL AXB(CPN,V,X(1),3,3,1)                                        SVSETUP  54
      VT=VTO                                                            SVSETUP  55
      ALT=ALTO                                                          SVSETUP  56
      RETURN                                                            SVSETUP  57
      END                                                               SVSETUP  58
```

```
      REAL FUNCTION TOUT(DMY)
C**  TOUT COMPUTES THE TIME AT WHICH THE NEXT OUTPUT IS REQUIRED
      COMMON /DTO/DTO(50)
      COMMON /SUPLE/SUPLE(9)
      EQUIVALENCE (SUPLE(1),T)
      EQUIVALENCE (SUPLE(6),ISEG)
      TOUT=(AINT(T/DTO(ISEG))+1.)*DTO(ISEG)
      RETURN
      END
```

```
      SUBROUTINE TSETUP1(TDONE)                                              TSETUP1     2
                                                                             TSETUP1     3
C**   PRIOR TO EACH VERTICAL TURN, TSETUP1 COMPUTES THE TIME AT WHICH        TSETUP1     4
C**   THE CHANGE IN PITCH ANGLE WILL EQUAL "PITCH".  IF AND WHEN SUCH        TSETUP1     5
C**   TIME IS REACHED, THE TURN IS COMPLETE AND THE VERTICAL TURN            TSETUP1     6
C**   ACCELERATION IS SWITCHED OFF IN SUBROUTINE FLTPATH.                    TSETUP1     7
                                                                             TSETUP1     8
      COMMON /PITCH/PITCH(50)                                                TSETUP1     9
      COMMON /PACC/PACC(50)                                                  TSETUP1    10
      COMMON /SUPLE/SUPLE(9)                                                 TSETUP1    11
      COMMON /STATE/STATE(23)                                                TSETUP1    12
      COMMON /TACC/TACC(50)                                                  TSETUP1    13
                                                                             TSETUP1    14
      EQUIVALENCE (SUPLE(3),TI)                                              TSETUP1    15
      EQUIVALENCE (SUPLE(6),ISEG)                                            TSETUP1    16
      EQUIVALENCE (STATE(4),VT)                                              TSETUP1    17
                                                                             TSETUP1    18
      IF (PACC(ISEG).EQ.C.) GO TO 10                                         TSETUP1    19
C                                                                            TSETUP1    20
C            ACCELERATED PATH MOTION                                         TSETUP1    21
      DT=VT*(EXP(PACC(ISEG)*ABS(PITCH(ISEG))/TACC(ISEG))-1.)/PACC(ISEG)      TSETUP1    22
      TDONE=TI+DT                                                            TSETUP1    23
      RETURN                                                                 TSETUP1    24
C                                                                            TSETUP1    25
C            UNACCELERATED PATH MOTION                                       TSETUP1    26
   10 DT=VT*ABS(PITCH(ISEG))/TACC(ISEG)                                      TSETUP1    27
      TDONE=TI+DT                                                            TSETUP1    28
      RETURN                                                                 TSETUP1    29
      END                                                                    TSETUP1    30
```

```
      SUBROUTINE TSETUP2(TOFF,TON,TDONE)

C**   PRIOR TO EACH HORIZONTAL TURN, TSETUP2 COMPUTES THE TIME AT WHICH
C**   ROLL-UP SHOULD CEASE (TOFF=T1), THE TIME AT WHICH ROLL-DOWN
C**   SHOULD BEGIN (TON=T2), AND THE TIME AT WHICH ROLL WILL BE
C**   RETURNED TO ZERO (TDONE).

      COMMON /HEAD/HEAD(50)
      COMMON /PACC/PACC(50)
      COMMON /PRBLK/PRBLK(13)
      COMMON /SEGLNT/SEGLNT(50)
      COMMON /STATE/STATE(23)
      COMMON /SUPLE/SUPLE(9)
      COMMON /TACC/TACC(50)

      EQUIVALENCE (PRBLK(13),ROLRATE)
      EQUIVALENCE (STATE(4),VT)
      EQUIVALENCE (SUPLE(2),TF)
      EQUIVALENCE (SUPLE(3),TI)
      EQUIVALENCE (SUPLE(5),ISEG)

      DIMENSION YR(2),YI(2)

      DT=SEGLNT(ISEG)
      CY=COS(ETAY(DMY))
      T1=ATAN(TACC(ISEG)/32.2/CY)/ROLRATE
      T2LEST1=0.
      IF ((2.*T1).LT.DT) T2LEST1=DT-2.*T1
      CALL YAWCHG(T1,T2LEST1,DYAW1,DYAW)
      IF (DYAW.GE.ABS(HEAD(ISEG))) GO TO 10
      IF (DYAW.LT.ABS(HEAD(ISEG))) GO TO 20
     1IF (DYAW1.GE.ABS(HEAD(ISEG))) GO TO 20
C
C        CASE 1  -  MAX ROLL REACHED AND TURN COMPLETED
   10 TOFF=TI+T1
      TON=0.
      VTDOT=PACC(ISEG)
      B0=32.2*CY*ALOG(COS(ROLRATE*T1))/ROLRATE
      B1=ABS(HEAD(ISEG))*CY
      B2=VT+VTDOT*T1/2.
      A=VTDOT*(0.5*B1*VTDOT+0.5*B0*VTDOT/B2-TACC(ISEG))
      B=1.5*B1*B2*VTDOT+2.*B0*VTDOT+TACC(ISEG)*(VTDOT*T1-B2)
      C=B2*(B1*B2+2.*B0+TACC(ISEG)*T1)
      CALL QUADRT(A,B,C,IWARN,YR,YI)
      IF (IWARN.EQ.1) WRITE (6,100)
      IF (IWARN.EQ.1) STOP
      IF (YR(1).GE.T1 .AND. YR(1).LE.DT) TON=TI+YR(1)
      IF (YR(2).GE.T1 .AND. YR(2).LE.DT) TON=TI+YR(2)
      IF (TON.EQ.0.) WRITE (6,101)
      IF (TON.EQ.0.) STOP
      TDONE=TON+(TOFF-TI)
      RETURN
C
C        CASE 2  -  MAX ROLL NOT REACHED BUT TURN COMPLETED
   20 TOFF=0.
      A=-40.7*ROLRATE*ROLRATE
      B=ROLRATE*ABS(HEAD(ISEG))*PACC(ISEG)
```

```
      C=ROLRATE*ABS(HEAD(ISEG))*VT                                      TSETUP2     59
      CALL QUADRT(A,B,C,IWARN,YR,YI)                                    TSETUP2     60
      IF (IWARN.EQ.1) WRITE (6,100)                                     TSETUP2     61
      IF (IWARN.EQ.1) STOP                                              TSETUP2     62
      IF (YR(1).GT.0.  .AND. YR(1).LE.DT) TOFF=TON=TI+YR(1)             TSETUP2     63
      IF (YR(2).GT.0.  .AND. YR(2).LE.DT) TOFF=TON=TI+YR(2)             TSETUP2     64
      IF (TOFF.EQ.0.) WRITE(6,102)                                      TSETUP2     65
      IF (TOFF.EQ.0.) STOP                                              TSETUP2     66
      IF (TOFF.GT.TF) TOFF=TON=TI+DT/2.                                 TSETUP2     67
      TDONE=TOFF+(TOFF-TI)                                              TSETUP2     68
      RETURN                                                            TSETUP2     69
   30 IF (T2LEST1.EQ.0.) GO TO 40                                       TSETUP2     70
C                                                                       TSETUP2     71
C            CASE C  -  MAX ROLL REACHED BUT TURN NOT COMPLETED         TSETUP2     72
      TOFF=TI+T1                                                        TSETUP2     73
      TON=TI+DT-T1                                                      TSETUP2     74
      TDONE=TF                                                          TSETUP2     75
      RETURN                                                            TSETUP2     76
C                                                                       TSETUP2     77
C            CASE D  -  MAX ROLL NOT REACHED AND TURN NOT COMPLETED     TSETUP2     78
   40 TOFF=TON=TI+DT/2.                                                 TSETUP2     79
      TDONE=TF                                                          TSETUP2     80
      RETURN                                                            TSETUP2     81
  100 FORMAT(T2,*TSETUP2 MESSAGE - IWARN=1. PROGRAM TERMINATED.*)       TSETUP2     82
  101 FORMAT(T2,*TSETUP2 MESSAGE - CASE A FAILURE. PROGRAM TERMINATED*) TSETUP2     83
  102 FORMAT(T2,*TSETUP2 MESSAGE - CASE B FAILURE. PROGRAM TERMINATED*) TSETUP2     84
      END                                                               TSETUP2     85
```

```
      SUBROUTINE VALDATA(NSEGT)
C**   VALDATA PERFORMS A RANGE CHECK ON ALL INPUT PARAMETERS THAT HAVE A
C**   RESTRICTED USEFUL RANGE.  IF ANY ARE OUT OF RANGE, AN INFORMATIVE
C**   MESSAGE IS PRINTED ABOUT EACH AND THEN THE RUN IS TERMINATED.

      COMMON /DTO/DTO(50)         /ERROR/ERROR(50)      /HEAD/HEAD(50)
      COMMON /HMAX/HMAX(50)       /HMIN/HMIN(50)
      COMMON /MODE/MODE(50)       /NPATH/NPATH(50)      /PITCH/PITCH(50)
      COMMON /PRBLK/PRBLK(13)     /SEGLNT/SEGLNT(50)
      COMMON /TACC/TACC(50)       /TURN/TURN(50)

      EQUIVALENCE (PRBLK(1),LLMECH)
      EQUIVALENCE (PRBLK(3),VTO)
      EQUIVALENCE (PRBLK(4),PHEADO)
      EQUIVALENCE (PRBLK(5),PPITCHO)
      EQUIVALENCE (PRBLK(6),ALFAO)
      EQUIVALENCE (PRBLK(7),LATO)
      EQUIVALENCE (PRBLK(8),LONO)
      EQUIVALENCE (PRBLK(13),ROLRATE)

      INTEGER TURN
      REAL LATO,LONO
      DIMENSION FINMESS(18),IERR(21)

      DATA IERR,ISTOP,HALFPI,PI/22*0,90.,180./
      DATA (FINMESS(I),I=1,18)/
     1 10H  NSEGT    ,10H  LLMECH   ,10H  VTO      ,10H  PHEADO   ,
     2 10H  PPITCHO  ,10H  ALFAO    ,10H  LATO     ,10H  LONO     ,
     3 10H  ROLRATE  ,10H  SEGLNT   ,10H  TURN     ,10H  NPATH    ,
     4 10H  TACC     ,10H  DTO      ,10H  MODE     ,10H  ERROR    ,
     5 10H  HMAX     ,10H  HMIN     /
C
C             RANGE-CHECK PRDATA
      IF (NSEGT.LT.1 .OR. NSEGT.GT.50) IERR(1)=1
      IF (LLMECH.LT.1 .OR. LLMECH.GT.4) IERR(2)=1
      IF (VTO.LT.0.) IERR(3)=1
      IF (PHEADO.LT.-PI .OR. PHEADO.GT.PI) IERR(4)=1
      IF (PPITCHO.LT.-HALFPI .OR. PPITCHO.GT.HALFPI) IERR(5)=1
      IF (ALFAO.LT.-PI .OR. ALFAO.GT.PI) IERR(6)=1
      IF (LATO.LE.-HALFPI .OR. LATO.GE.HALFPI) IERR(7)=1
      IF (LONO.LT.-PI .OR. LONO.GT.PI) IERR(8)=1
      IF (ROLRATE.LE.0.) IERR(9)=1
C
C             RANGE-CHECK PASDATA
      DO 10 I=1,50
         IF (SEGLNT(I).LT.0.) IERR(10)=1
         IF (TURN(I).LT.1 .OR. TURN(I).GT.4) IERR(11)=1
         IF (NPATH(I).LT.1 .OR. NPATH(I).GT.2) IERR(12)=1
         IF (TACC(I).LT.0.) IERR(13)=1
         IF (DTO(I).LE.0.) IERR(14)=1
         IF (MODE(I).LT.0 .OR. MODE(I).GT.1) IERR(15)=1
         IF (ERROR(I).LE.0.) IERR(16)=1
         IF (HMAX(I).LE.0.) IERR(17)=1
         IF (HMIN(I).LE.0.) IERR(18)=1
         IF (TURN(I) NE.3) GO TO 5
```

```
          IF (HEAD(I).LE.-HALFPI .OR. HEAD(I).GF.HALFPI) IERR(19)=1        VALDATA  59
          IF (PITCH(I).EQ.0.) IERR(20)=1                                   VALDATA  60
    5     IF (TURN(I).NE.1 .AND. TURN(I).NE.2) GO TO 10                     VALDATA  61
          IF (TACC(I).EQ.0.) IERR(21)=1                                    VALDATA  62
   10  CONTINUE                                                            VALDATA  63
C                                                                          VALDATA  64
C             PRINT MESSAGES IF REQUIRED                                   VALDATA  65
       DO 20 I=1,9                                                         VALDATA  66
          IF (IERR(I).EQ.0) GO TO 20                                       VALDATA  67
          WRITE (6,100) FINMESS(I)                                         VALDATA  68
          ISTOP=1                                                          VALDATA  69
   20  CONTINUE                                                            VALDATA  70
       DO 30 I=10,18                                                       VALDATA  71
          IF (IERR(I).EQ.0) GO TO 30                                       VALDATA  72
          WRITE (6,110) FINMESS(I)                                         VALDATA  73
          ISTOP=1                                                          VALDATA  74
   30  CONTINUE                                                            VALDATA  75
       IF (IERR(19).EQ.1) WRITE (6,120)                                    VALDATA  76
       IF (IERR(20).EQ.1) WRITE (6,130)                                    VALDATA  77
       IF (IERR(21).EQ.1) WRITE (6,140)                                    VALDATA  78
       IF (IERR(19).EQ.1 .OR. IERR(20).EQ.1 .OR. IERR(21).EQ.1) ISTOP=1   VALDATA  79
       IF (ISTOP.EQ.1) WRITE (6,150)                                       VALDATA  80
       IF (ISTOP.EQ.1) STOP                                                VALDATA  81
       RETURN                                                              VALDATA  82
  100  FORMAT(/T2,*THIS PRDATA PARAMETER IS OUT OF RANGE : *,A10)          VALDATA  83
  110  FORMAT(/T2,*AT LEAST ONE ELEMENT OF THIS PASDATA PARAMETER IS OUT   VALDATA  84
      1 OF RANGE : *,A10)                                                  VALDATA  85
  120  FORMAT(/T2,*THE HEADING VARIATION (HEAD) FOR ONE OF THE SINE HEADI VALDATA  86
      1NG CHANGE MANEUVERS IS GREATER THAN 90 DEGREES.*)                   VALDATA  87
  130  FORMAT(/T2,*THE OSCILLATION FREQUENCY (PITCH) FOR ONE OF THE SINE  VALDATA  88
      1HEADING CHANGE MANEUVERS IS 0.*)                                    VALDATA  89
  140  FORMAT(/T2,*TURN ACCELERATION (TACC) IS ZERO FOR SOME VERTICAL OR  VALDATA  90
      1HORIZONTAL TURN.*)                                                  VALDATA  91
  150  FORMAT(/T2,*THE ABOVE ERROR(S) COULD BE FATAL IN EXECUTION SO PROF VALDATA  92
      1GEN IS TERMINATED HERE.*)                                           VALDATA  93
       END                                                                 VALDATA  94
```

```
      SUBROUTINE VDOT(DVX,DVY,DVZ)

C**   VDOT COMPUTES THE DERIVATIVES OF THE EARTH FRAME
C**   VELOCITIES (VX,VY,VZ) AS COORDINATIZED IN THE NAV FRAME.

      COMMON /PACC/PACC(50)
      COMMON /STATE/STATE(23)
      COMMON /SUPLE/SUPLE(9)

      EQUIVALENCE (STATE(1),VX)
      EQUIVALENCE (STATE(2),VY)
      EQUIVALENCE (STATE(3),VZ)
      EQUIVALENCE (STATE(6),CPN11)
      EQUIVALENCE (STATE(7),CPN21)
      EQUIVALENCE (STATE(8),CPN31)
      EQUIVALENCE (SUPLE(6),ISEG)
      EQUIVALENCE (WPN(1),WPNX)
      EQUIVALENCE (WPN(2),WPNY)
      EQUIVALENCE (WPN(3),WPNZ)

      DIMENSION WPN(3)

      DVT=PACC(ISEG)
      CALL OMEGAPN(WPN)
      DVX=CPN11*DVT-WPNZ*VY+WPNY*VZ
      DVY=CPN21*DVT-WPNX*VZ+WPNZ*VX
      DVZ=CPN31*DVT-WPNY*VX+WPNX*VY
      RETURN
      END
```

```
FUNCTION VEAST      74/74    OPT=2                         FTN 4.5+414        05/11/76  13.22.47        PAGE    63

                REAL FUNCTION VEAST(DMY)                                          VEAST       2
                                                                                  VEAST       3
          C**   VEAST COMPUTES THE EAST COMPONENT OF VELOCITY                     VEAST       4
                                                                                  VEAST       5
                COMMON /STATE/STATE(23)                                           VEAST       6
                EQUIVALENCE (STATE(1),VX)                                         VEAST       7
                EQUIVALENCE (STATE(2),VY)                                         VEAST       8
                                                                                  VEAST       9
                A=ALFA(DMY)                                                       VEAST      10
                VEAST=-VX*SIN(A)-VY*COS(A)                                        VEAST      11
                RETURN                                                            VEAST      12
                END                                                               VEAST      13
```

SUBROUTINE YAWCHG   74/74   OPT=2   FTN 4.5+414   06/11/76 13.22.47   PAGE   64

```
      SUBROUTINE YAWCHG(T1,T2LEST1,DYAW1,DYAW)

C**   YAWCHG COMPUTES THE ANGLE THROUGH WHICH THE AIRCRAFT COULD YAW
C**   IF IT REMAINED IN A HORIZONTAL TURN FOR THE ENTIRE FLIGHT SEGMENT.

      COMMON /PACC/PACC(50)
      COMMON /PRBLK/PRBLK(13)
      COMMON /SEGLNT/SEGLNT(50)
      COMMON /STATE/STATE(23)
      COMMON /SUPLE/SUPLE(9)
      COMMON /TACC/TACC(50)

      EQUIVALENCE (PRBLK(13),ROLRATE)
      EQUIVALENCE (STATE(4),VT)
      EQUIVALENCE (SUPLE(6),ISEG)

      DTROLL=SEGLNT(ISEG)/2.
      IF (T2LEST1.GT.0.) DTROLL=T1
      T2=SEGLNT(ISEG)-DTROLL
      VTDOT=PACC(ISEG)
C
C             COMPUTE YAW CHANGE THAT OCCURS WHILE ROLLING
C             INTO AND OUT OF THE TURN.
      DYAW1=(-32.2*ALOG(COS(ROLRATE*DTROLL))/ROLRATE) *
     1       (1./(VT+VTDOT*DTROLL/2.) + 1./(VT+VTDOT*(T2+DTROLL/2.)))
C
C             COMPUTE YAW CHANGE THAT OCCURS WHILE HOLDING
C             CONSTANT ROLL ANGLE DURING THE TURN.
      VT1=VT+VTDOT*T1
      IF (VTDOT.EQ.0.) DYAW2=TACC(ISEG)*T2LEST1/(VT1*COS(ETAY(DMY)))
      IF (VTDOT.NE.0.) DYAW2=TACC(ISEG)*ALOG(1.+VTDOT*T2LEST1/VT1)/
     1   (VTDOT*COS(ETAY(DMY)))
C
C             TOTAL YAW CHANGE
      DYAW=DYAW1+DYAW2
      RETURN
      END
```

```
***********        IPU02A0    //// END  OF LIST ////
```

REFERENCES

1. T. O. Seppelin, "The Department of Defense World Geodetic System 1972", Defense Mapping Agency, Washington, D.C., May 1974.

2. G. L. Hosmer, Geodesy, Wiley and Sons, 1930.

3. J. C. Pinson, "Inertial Guidance for Cruise Vehicles," in Leondes, C. T. ed., Guidance and Control of Aerospace Vehicles, McGraw-Hill, New York, 1963.

4. W. A. Heiskanen and H. Moritz, Physical Geodesy, W. H. Freeman, 1967.

5. P. S. Maybeck, "Wander Azimuth Implementation Algorithm for a Strapdown Inertial System", AFFDL-TR-73-80, AD784752, Air Force Flight Dynamic Laboratory, WPAFB, Ohio, Oct 1973.

6. R. F. Osborn and I. J. Dotterer, "SAMUS,A Program for State Space Analysis of Multisensor Systems", Vol III, T70-428/201, Autonetics, Anaheim, California, May 1971.

7. B. W. Kernighan and P. J. Plauger, The Elements of Programming Style, McGraw-Hill, 1974.

☆ U. S. GOVERNMENT PRINTING OFFICE: 1976 — 657-001/140